

A Journey in Polyglot



Chris Bailey(@Chris_Bailey)
IBM Runtime Technologies

January 21st, 2017



Swift @ IBM



Swift @ IBM

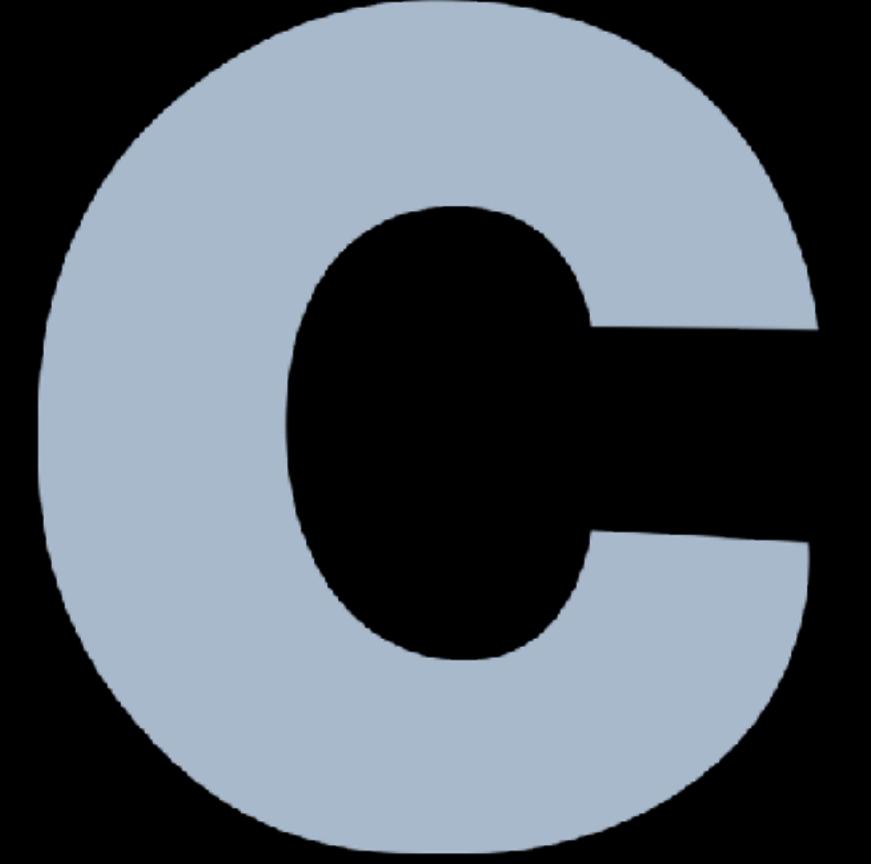


Swift @ IBM



Swift @ IBM

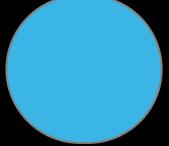
THE



PROGRAMMING
LANGUAGE



Runtime

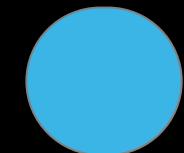
 Java Code

 C Code

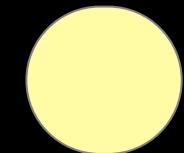


Runtime

Interpreter and JIT Compiler



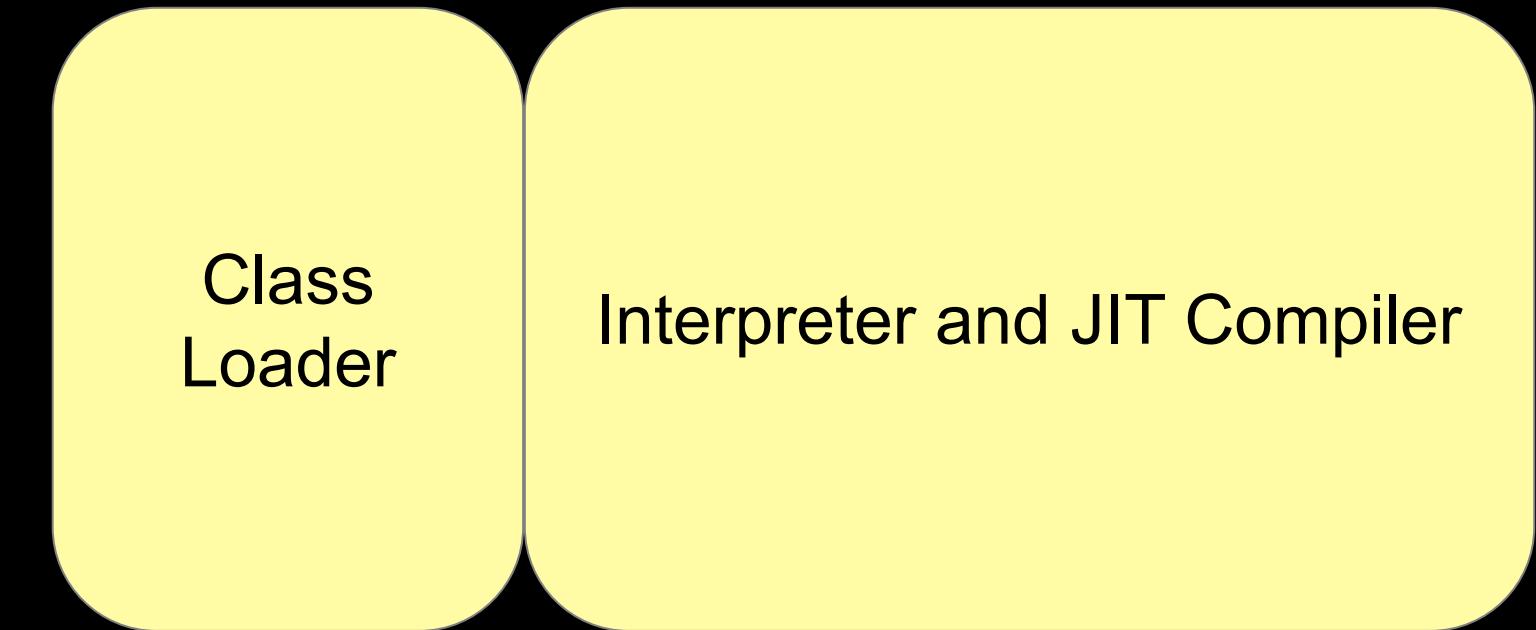
Java Code



C Code



Runtime

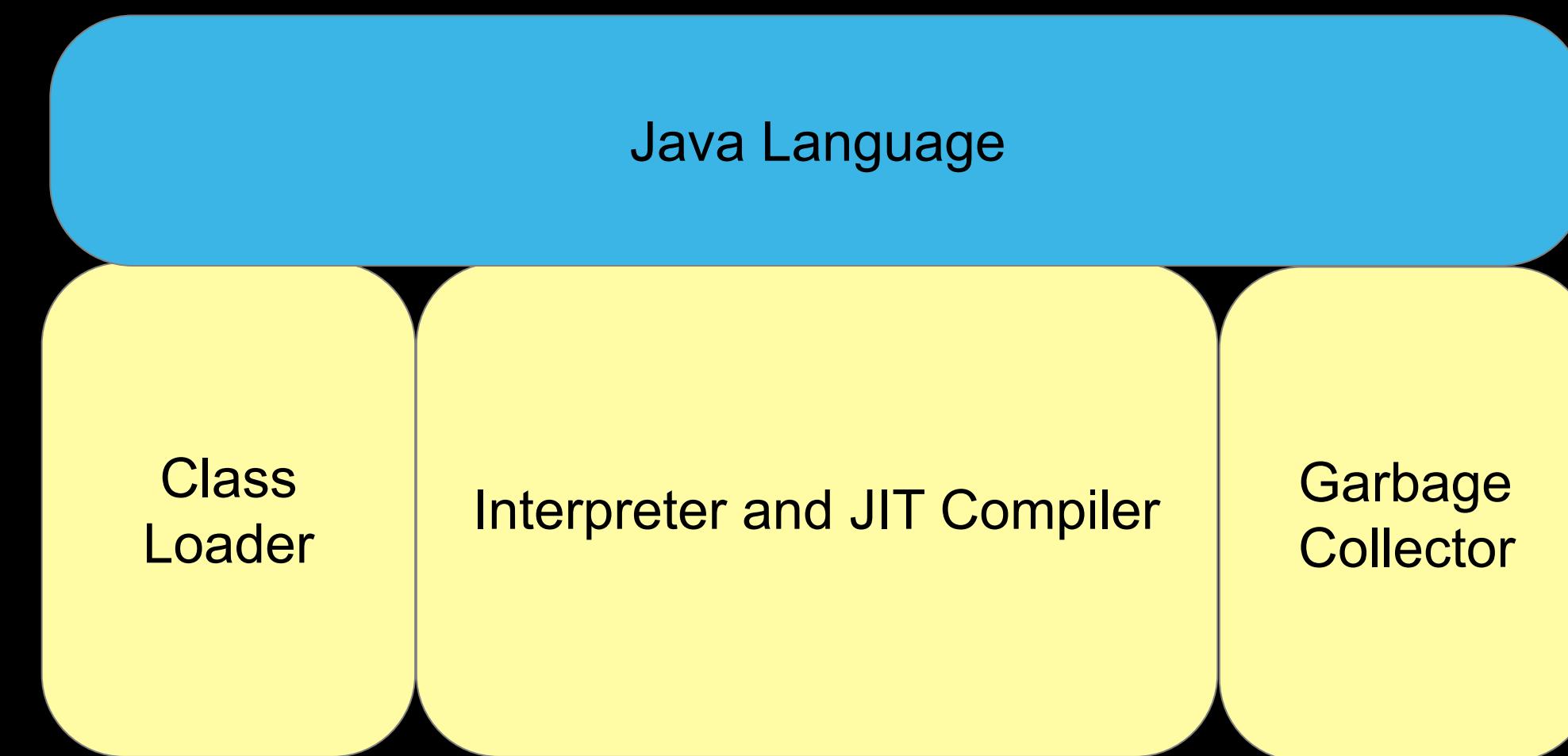


Java Code

C Code



Runtime

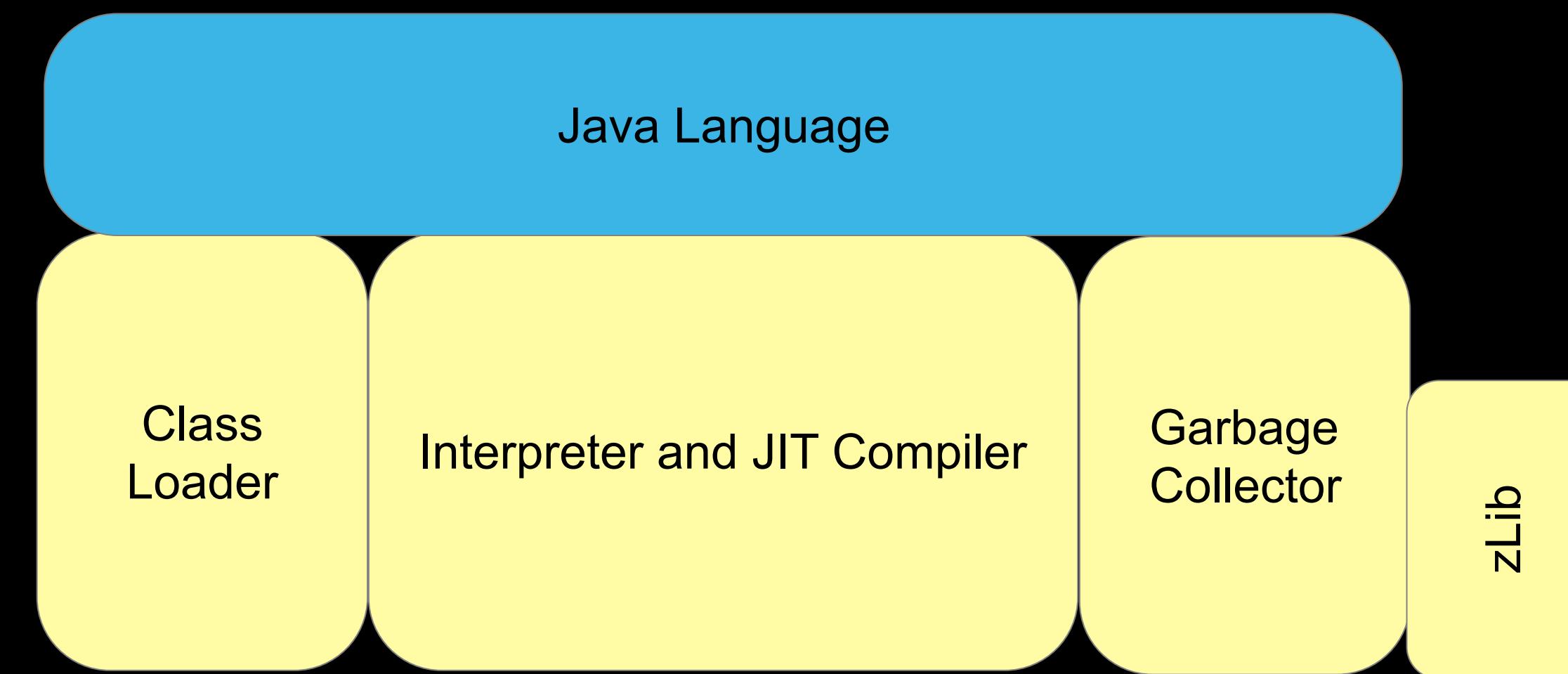


● Java Code

● C Code



Runtime

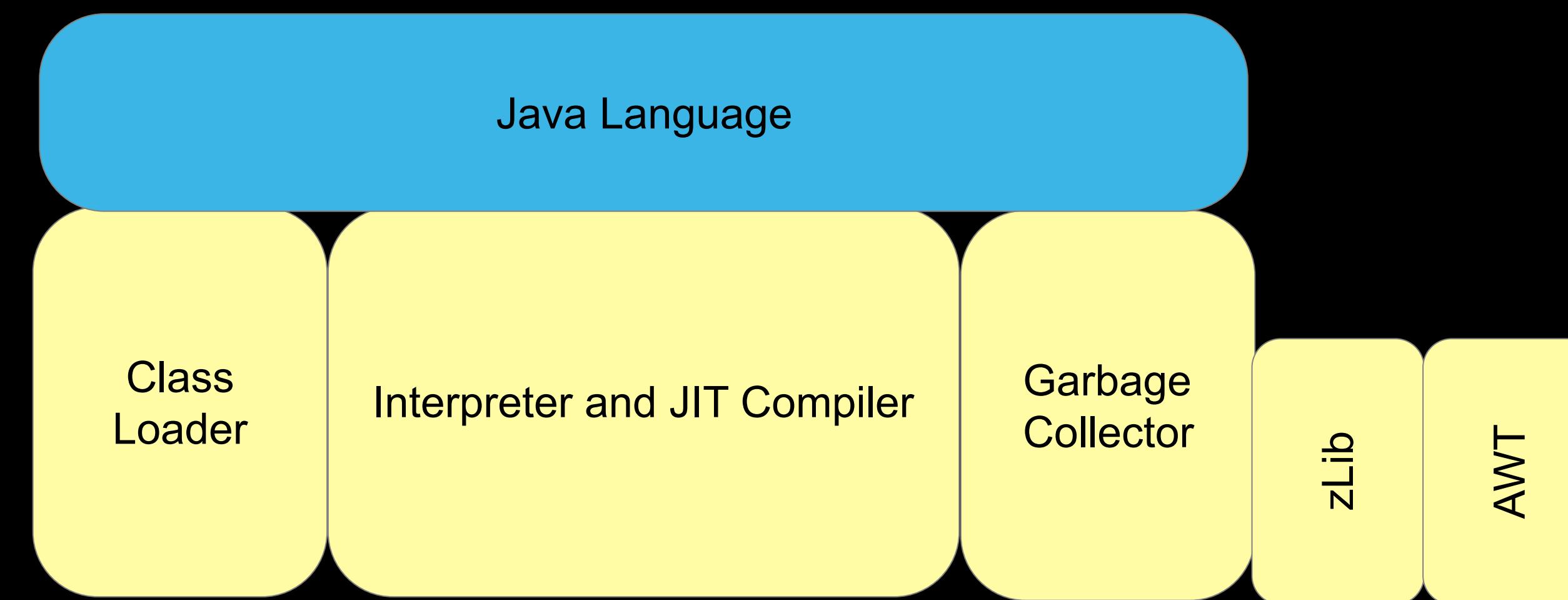


Java Code

C Code



Runtime

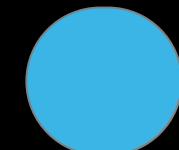
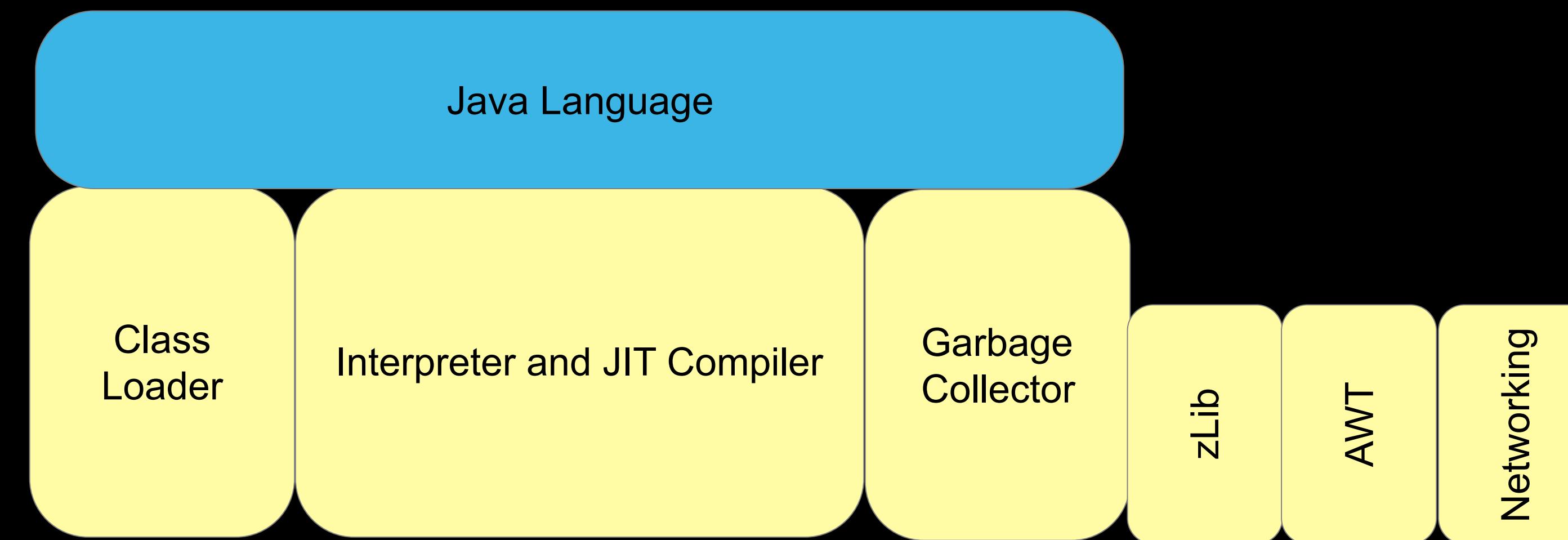


Java Code

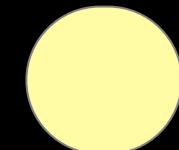
C Code



Runtime



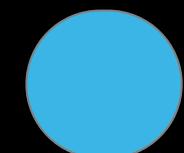
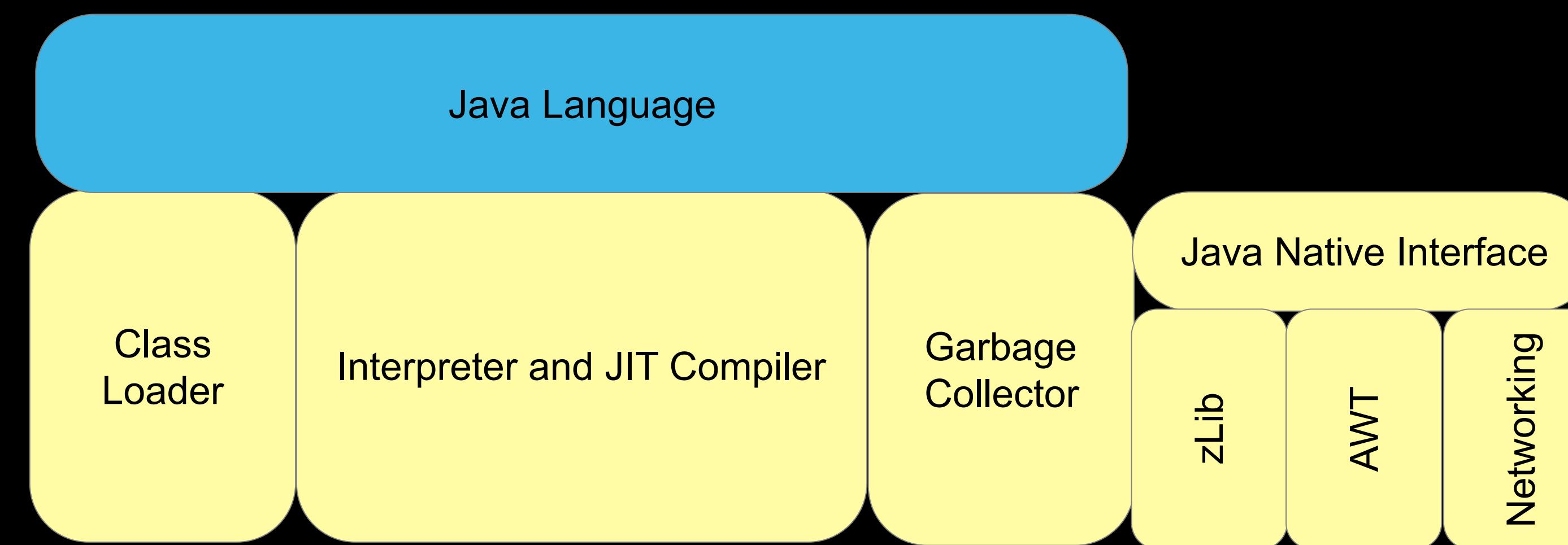
Java Code



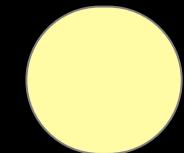
C Code



Runtime



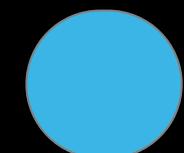
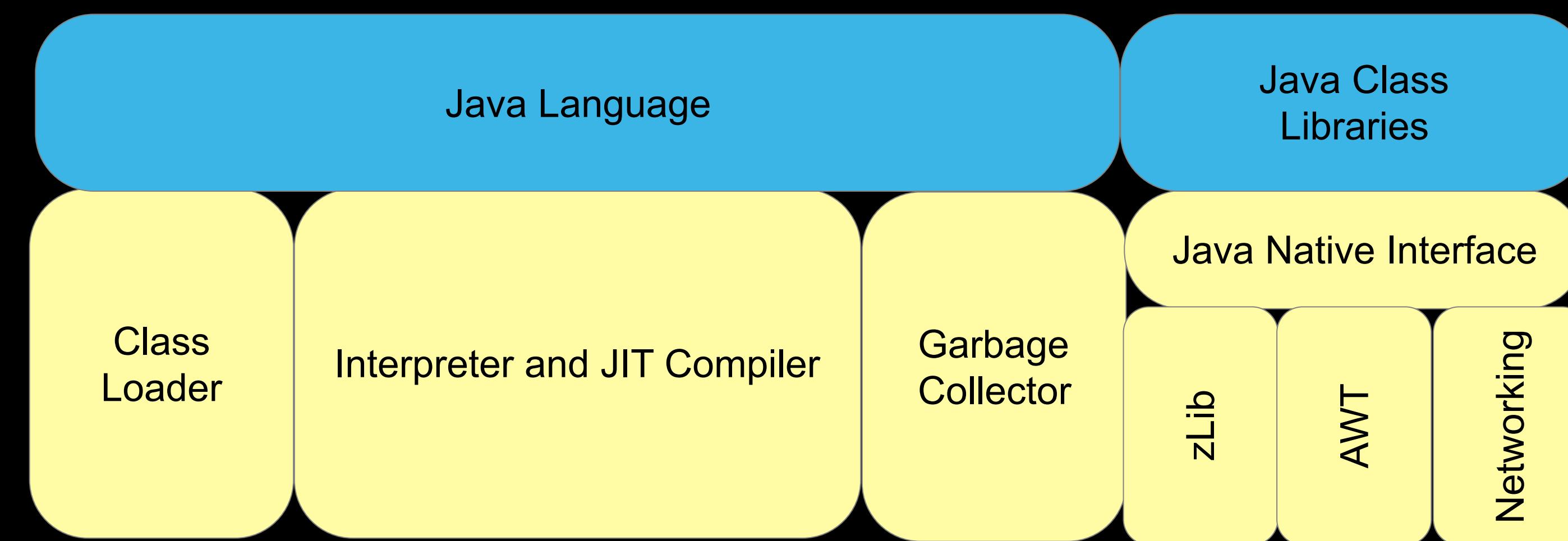
Java Code



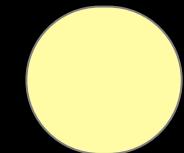
C Code



Runtime



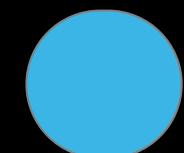
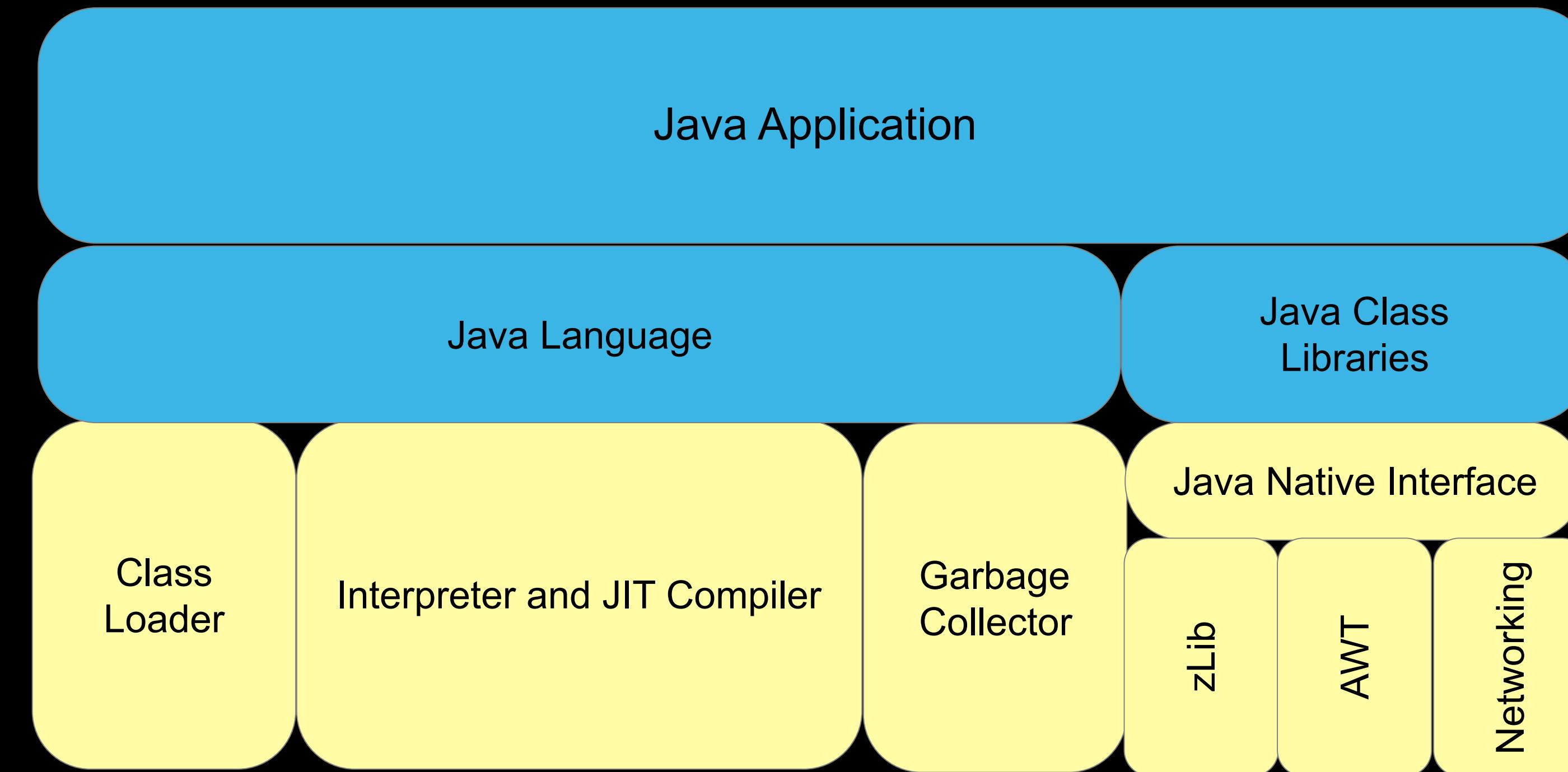
Java Code



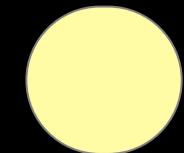
C Code



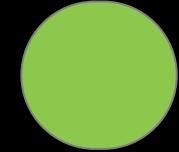
Runtime



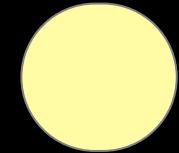
Java Code



C Code

 Runtime

JavaScript Code

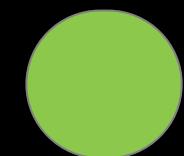


C Code

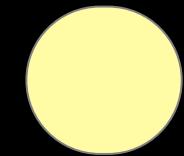


Runtime

Interpreter and JIT Compiler

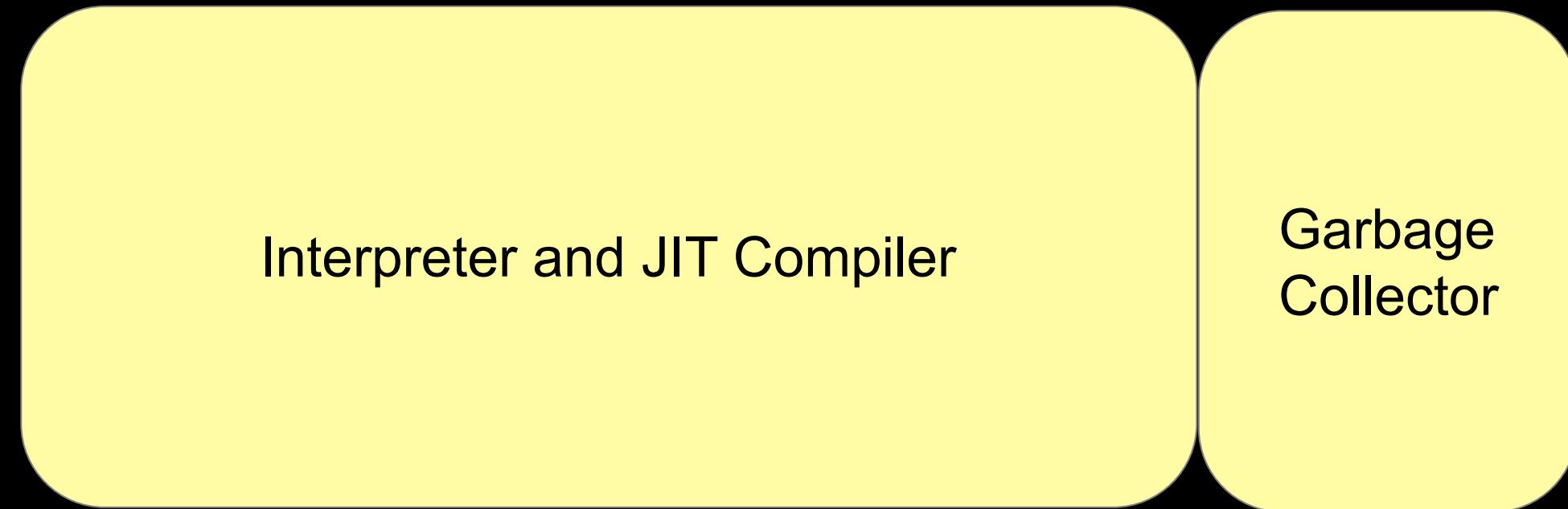


JavaScript Code



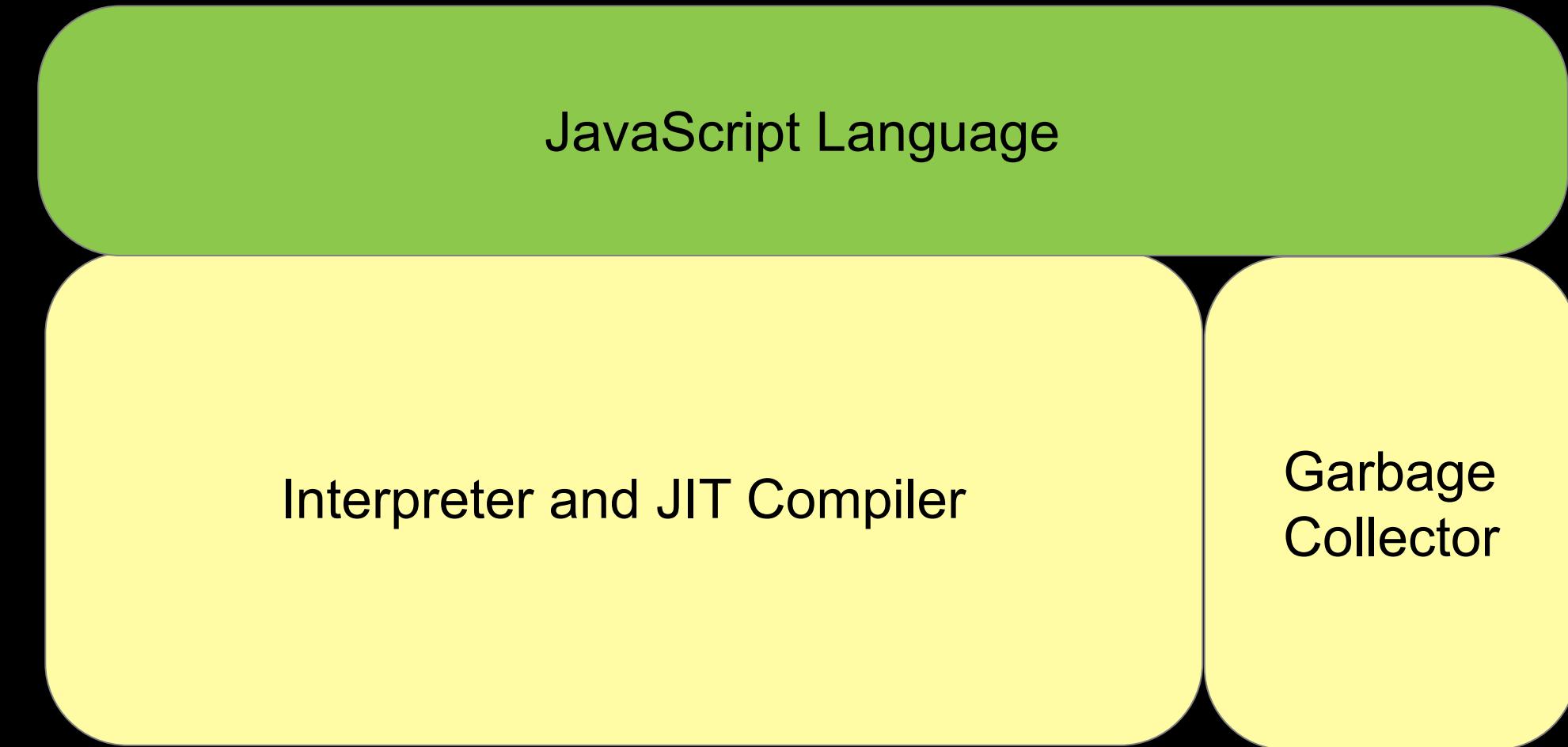
C Code

JS Runtime



- JavaScript Code
- C Code

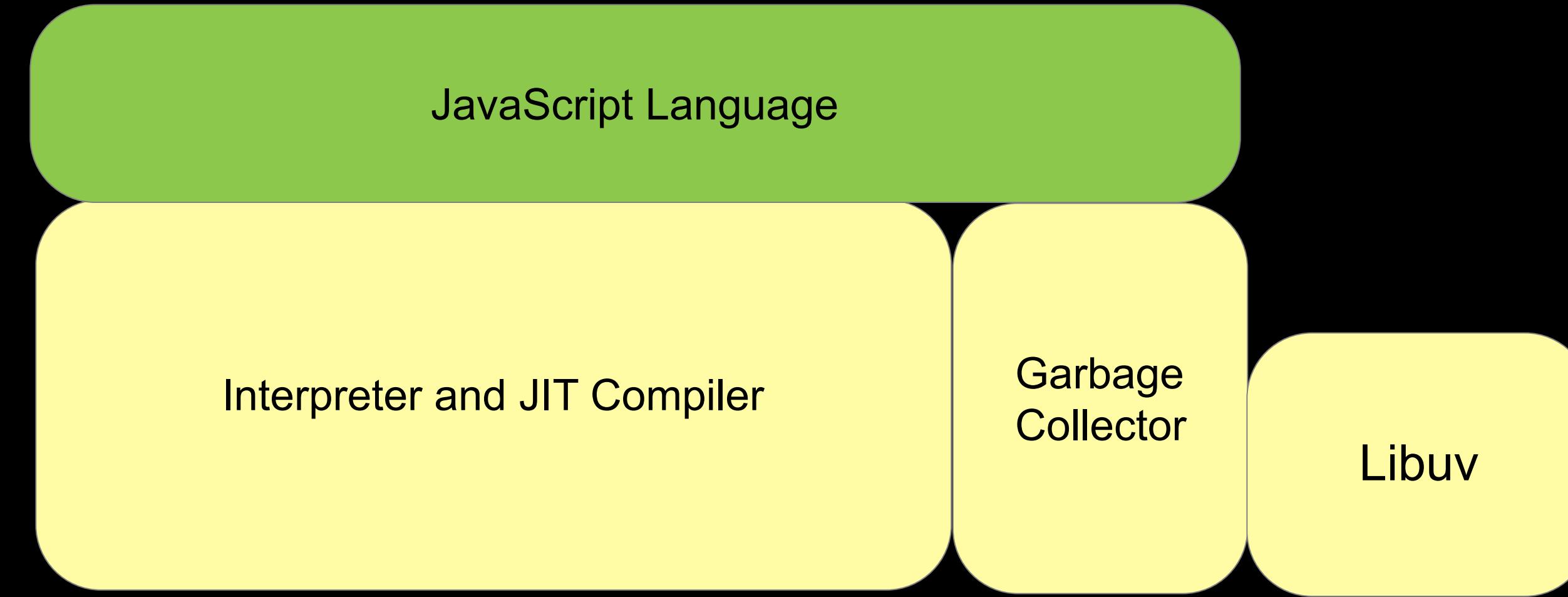
JS Runtime



● JavaScript Code

● C Code

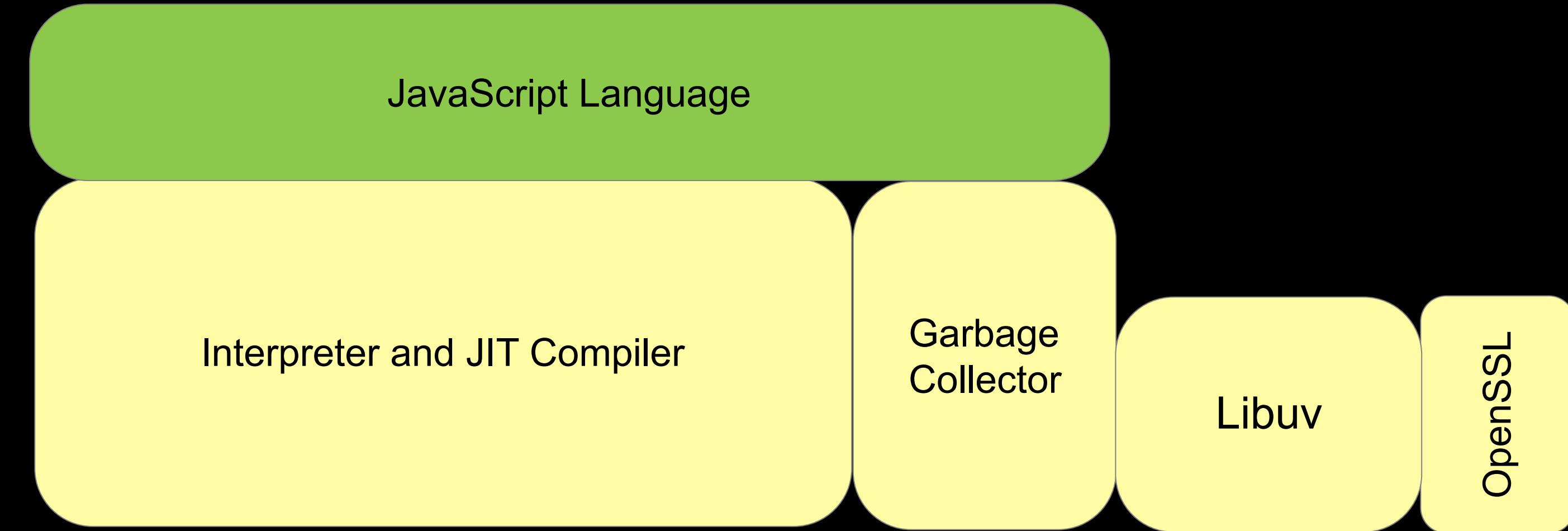
JS Runtime



JavaScript Code

C Code

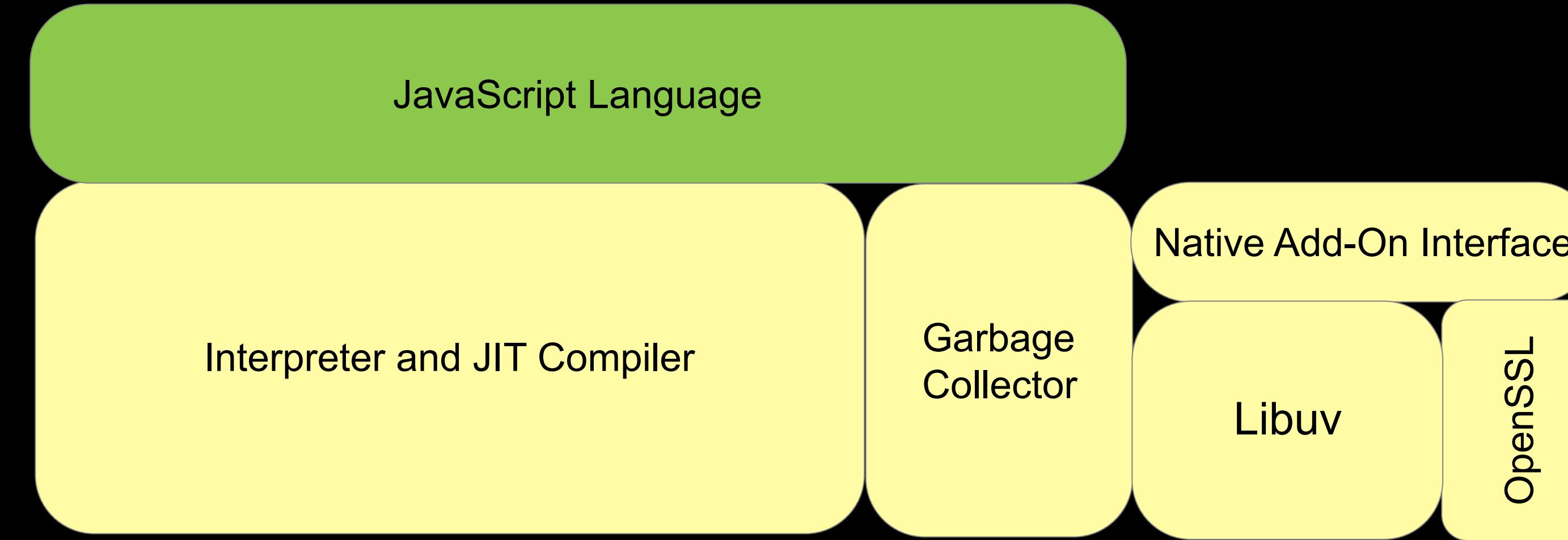
JS Runtime



● JavaScript Code

● C Code

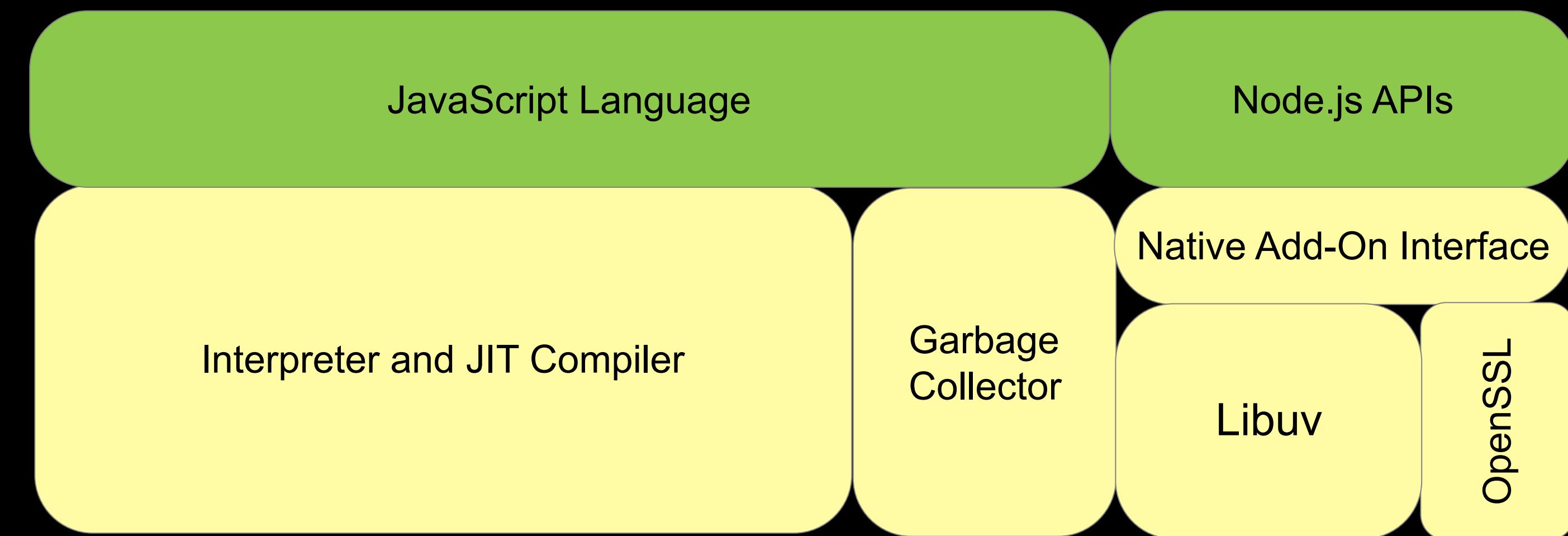
JS Runtime



JavaScript Code

C Code

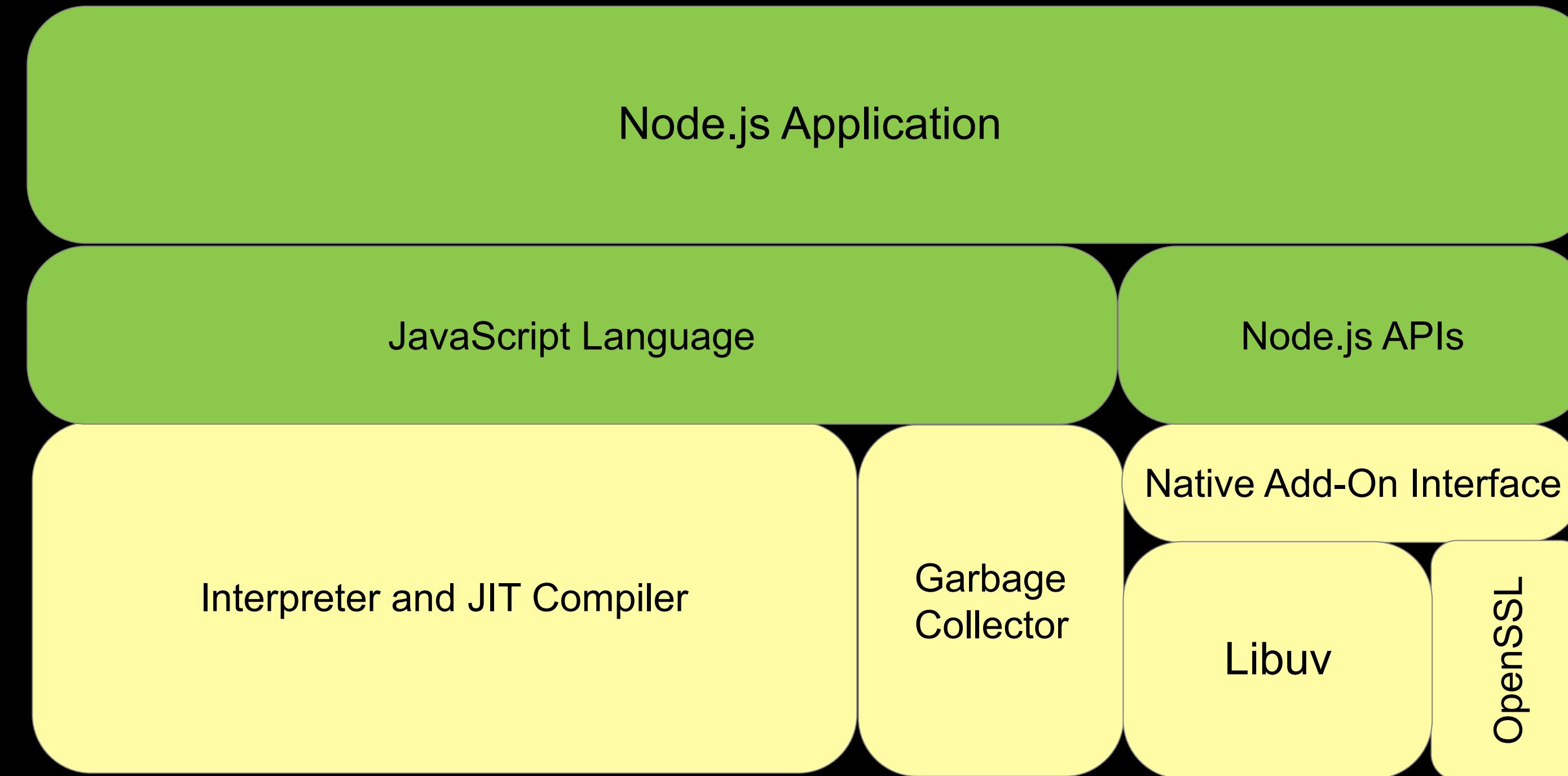
JS Runtime



● JavaScript Code

● C Code

JS Runtime

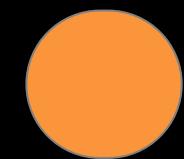


● JavaScript Code

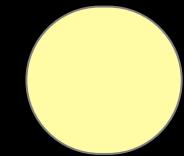
● C Code



Runtime



Swift Code

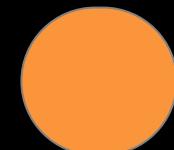


C Code

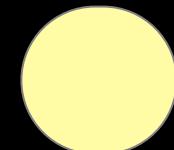


Runtime

LLVM Runtime and Compiler



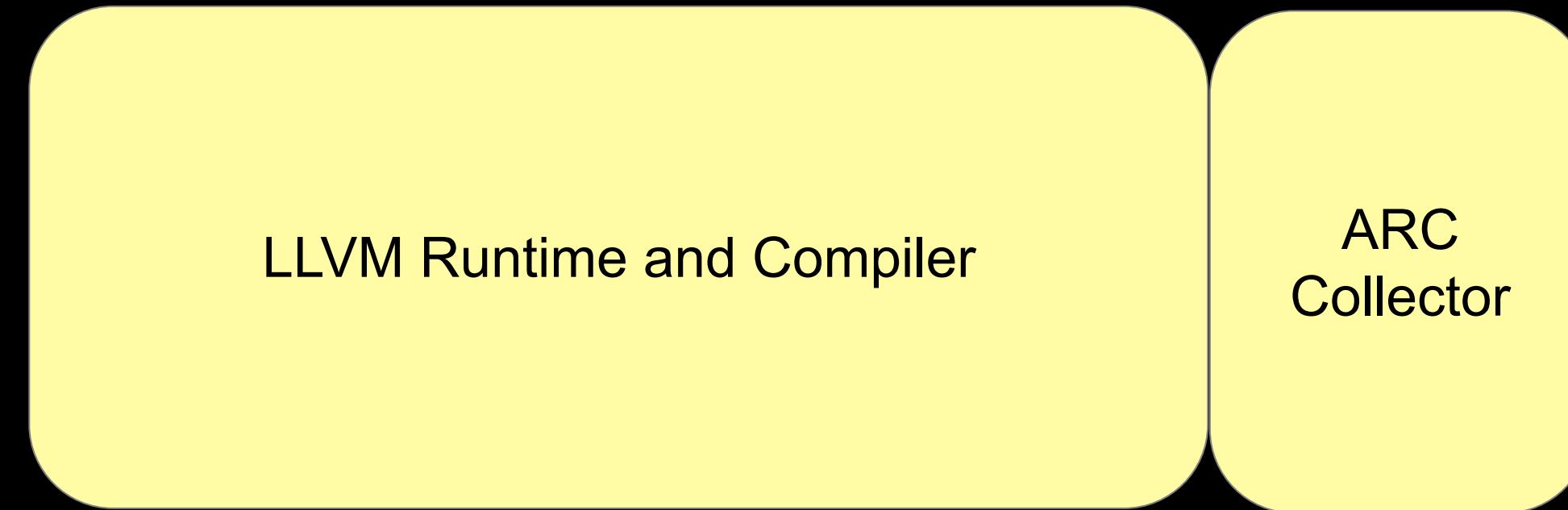
Swift Code



C Code



Runtime

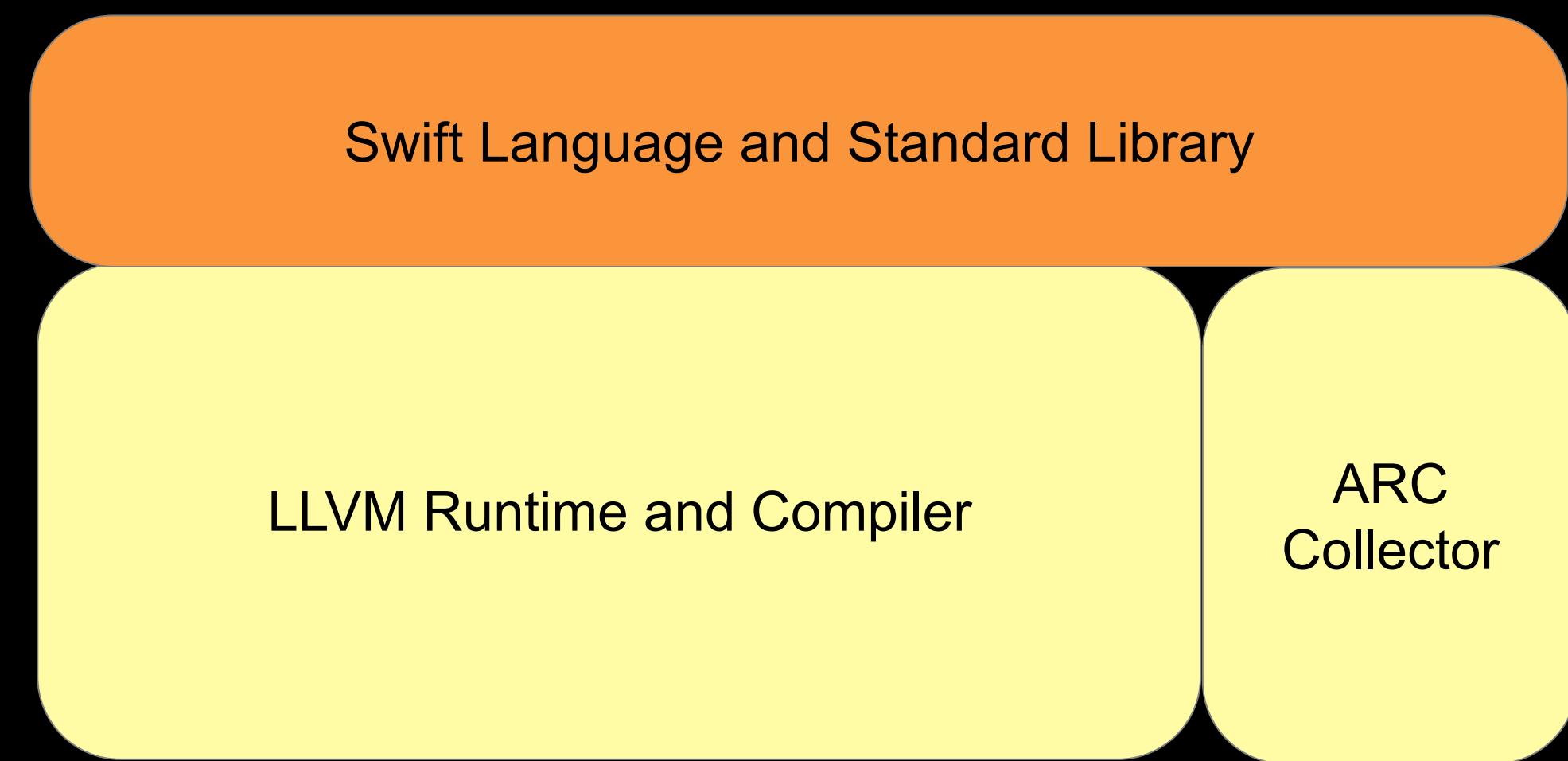


● Swift Code

● C Code



Runtime

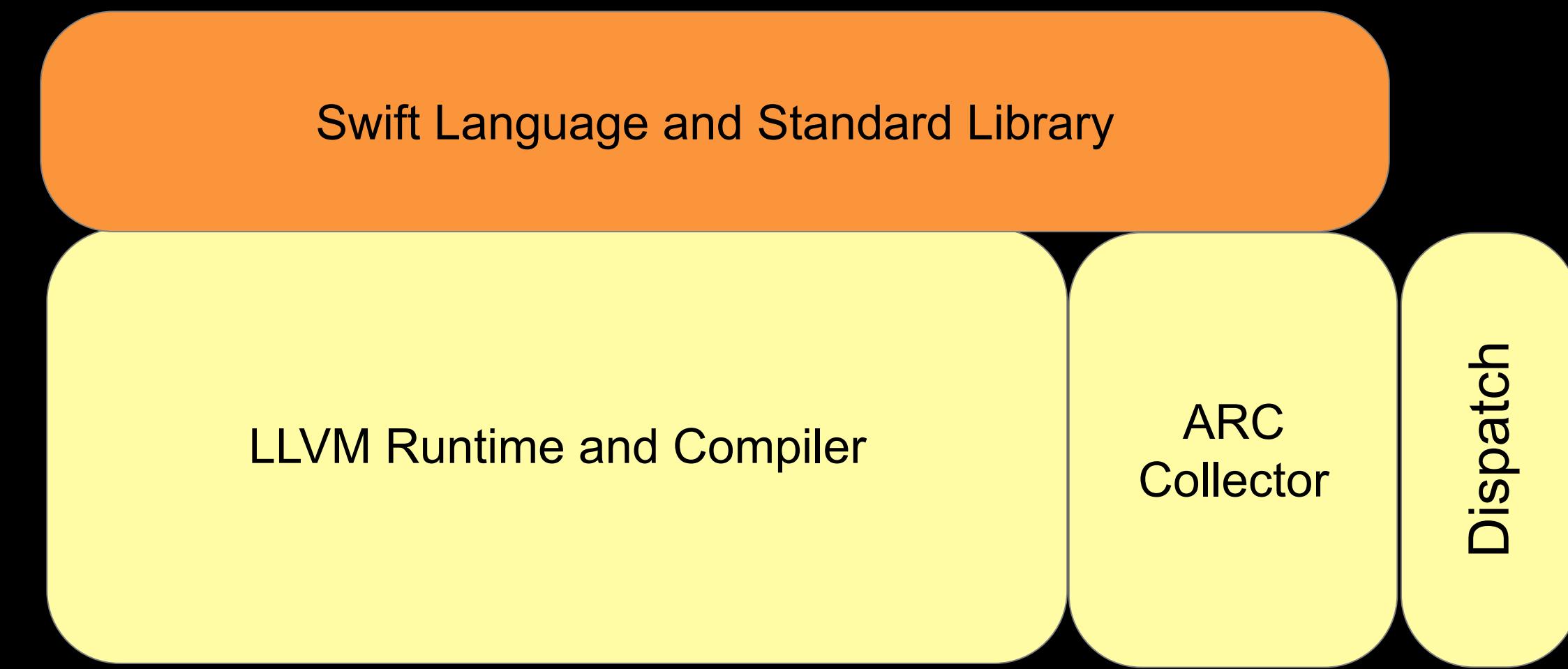


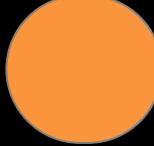
● Swift Code

● C Code



Runtime

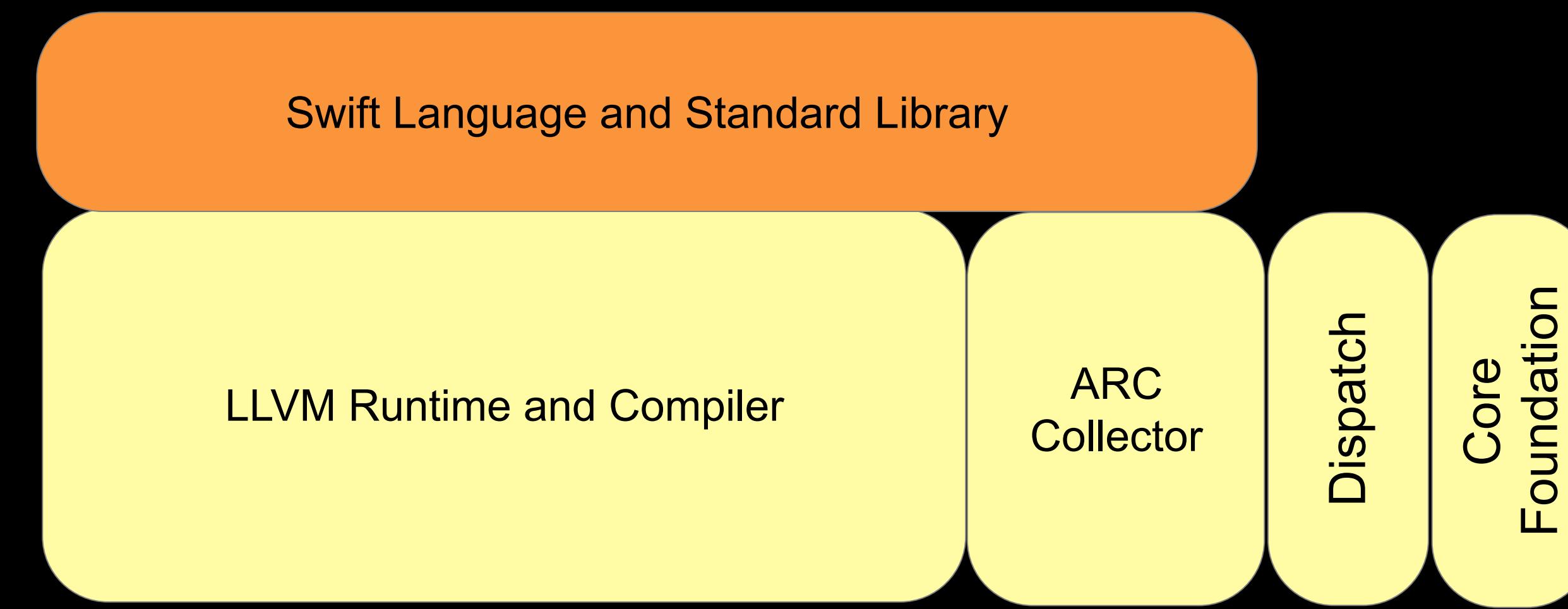


 Swift Code

 C Code



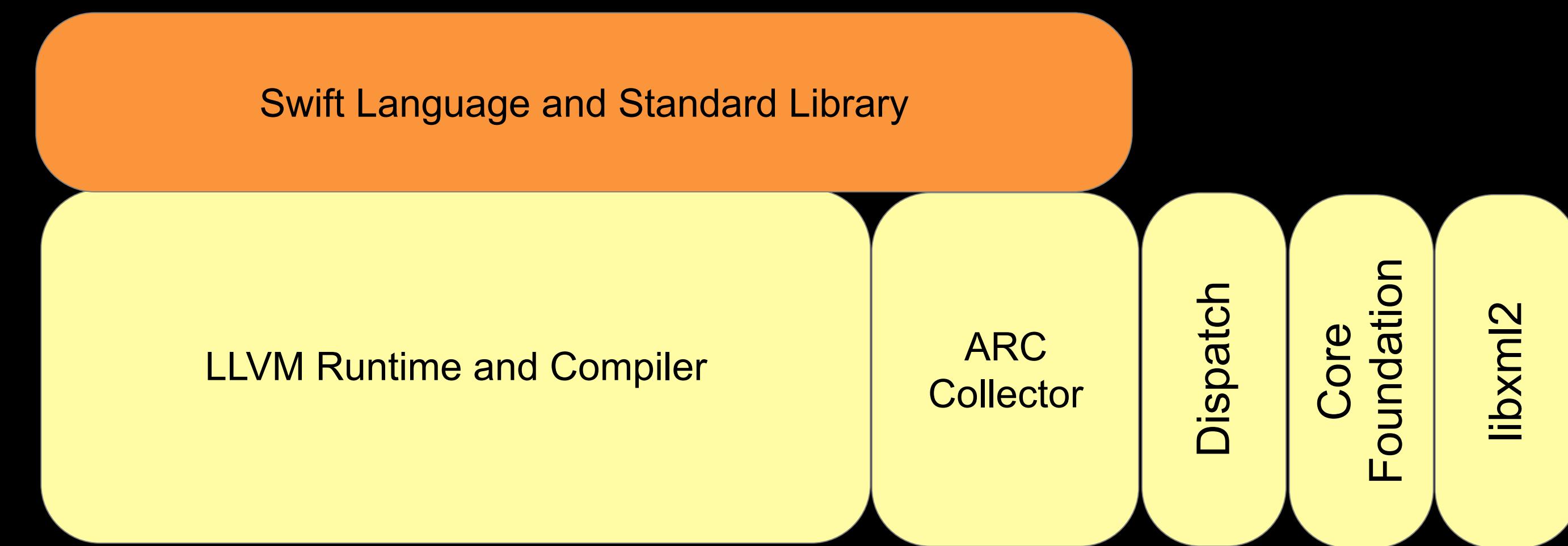
Runtime



- Swift Code
- C Code



Runtime

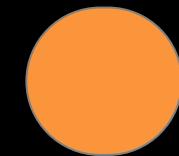
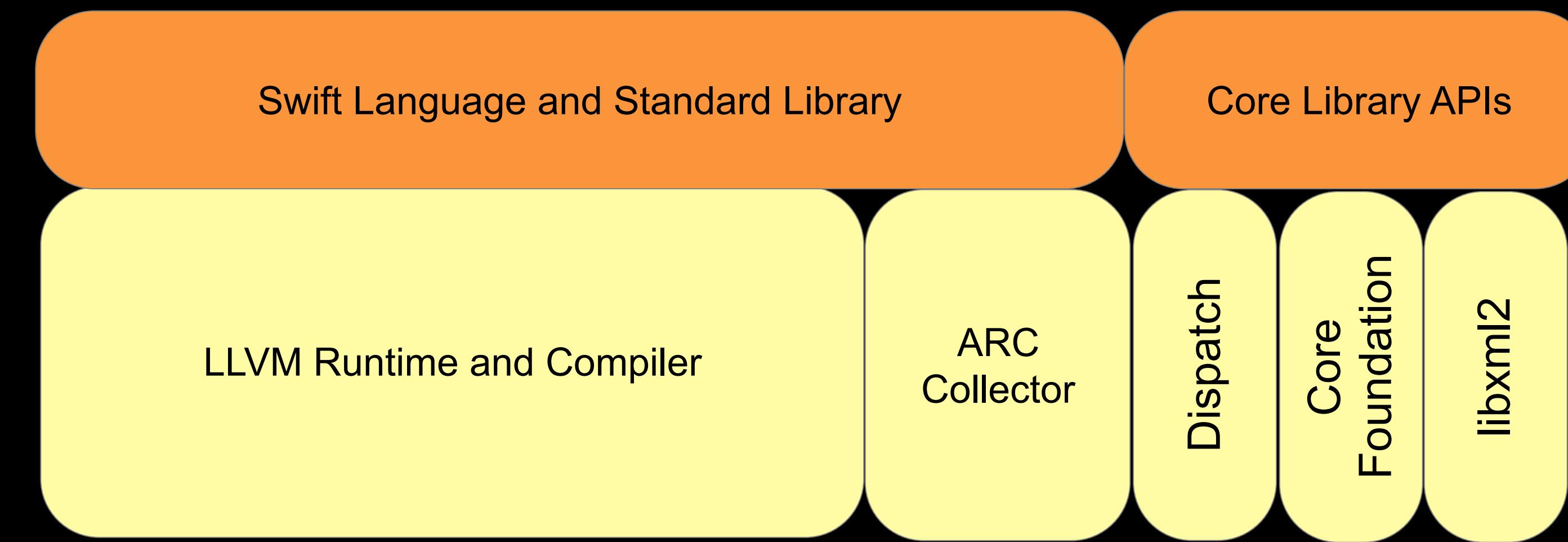


 Swift Code

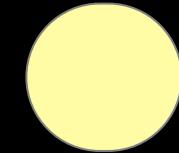
 C Code



Runtime



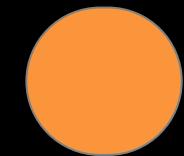
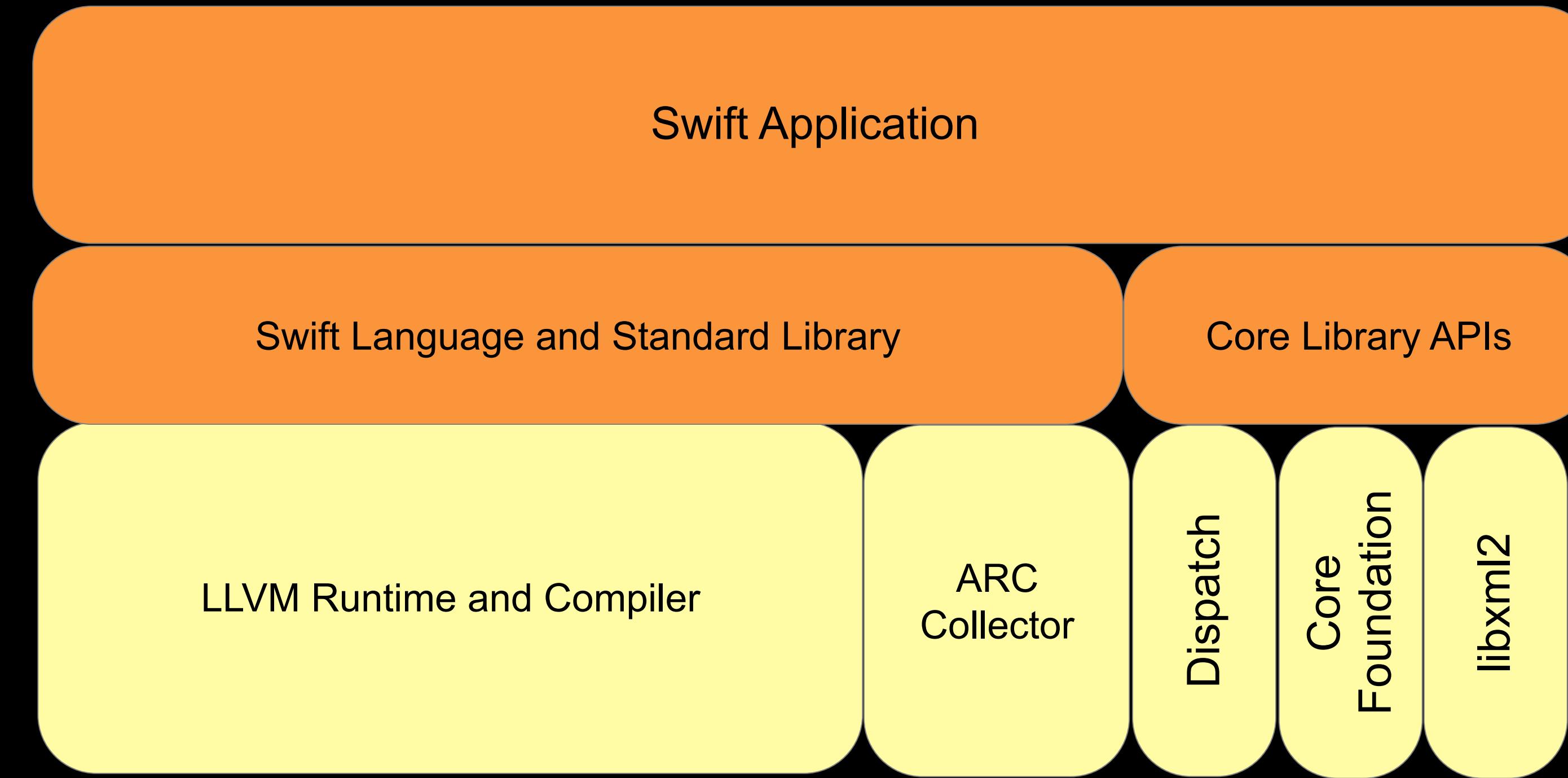
Swift Code



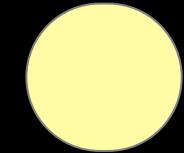
C Code



Runtime



Swift Code



C Code

Type Safety

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = 5;  
var b = 3;  
  
add(a, b);
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = 5  
let b = 3  
  
add(a, to: b)
```

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = 5;  
var b = 3;  
  
add(a, b);
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = 5  
let b = 3  
  
add(a, to: b)  
  
> swiftc main.swift
```

Safety



```
var add = function (a, b) {  
    console.log(a + b);  
}
```

```
var a = 5;  
var b = 3;
```

```
add(a, b);
```

```
> node app.js
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}
```

```
let a = 5  
let b = 3
```

```
add(a, to: b)
```

```
> swiftc main.swift  
> main
```

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = 5;  
var b = 3;  
  
add(a, b);
```

```
> node app.js  
  
> 8
```

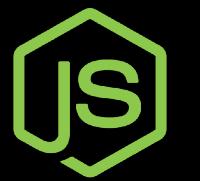


Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = 5  
let b = 3  
  
add(a, to: b)
```

```
> swiftc main.swift  
> main  
  
> 8
```

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = '5';  
var b = 3;  
  
add(a, b);
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = "5"  
let b = 3  
  
add(a, to: b)
```

Safety



```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = '5';  
var b = 3;  
  
add(a, b);
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = "5"  
let b = 3  
  
add(a, to: b)  
  
> swiftc main.swift
```

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = '5';  
var b = 3;  
  
add(a, b);
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = "5"  
let b = 3  
  
add(a, to: b)  
  
> swiftc main.swift  
  
> Error line 6: Cannot convert value of  
type 'String' into argument of type  
'Int'
```

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = '5';  
var b = 3;  
  
add(a, b);
```

```
> node app.js
```



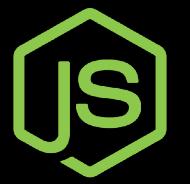
Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = "5"  
let b = 3  
  
add(a, to: b)
```

```
> swiftc main.swift
```

```
> Error line 6: Cannot convert value of  
type 'String' into argument of type  
'Int'
```

Safety



Node.js

```
var add = function (a, b) {  
    console.log(a + b);  
}  
  
var a = '5';  
var b = 3;  
  
add(a, b);
```

```
> node app.js  
> 53
```



Swift

```
func add(_ a: Int, to b: Int) -> Void {  
    print(a + b)  
}  
  
let a = "5"  
let b = 3  
  
add(a, to: b)
```

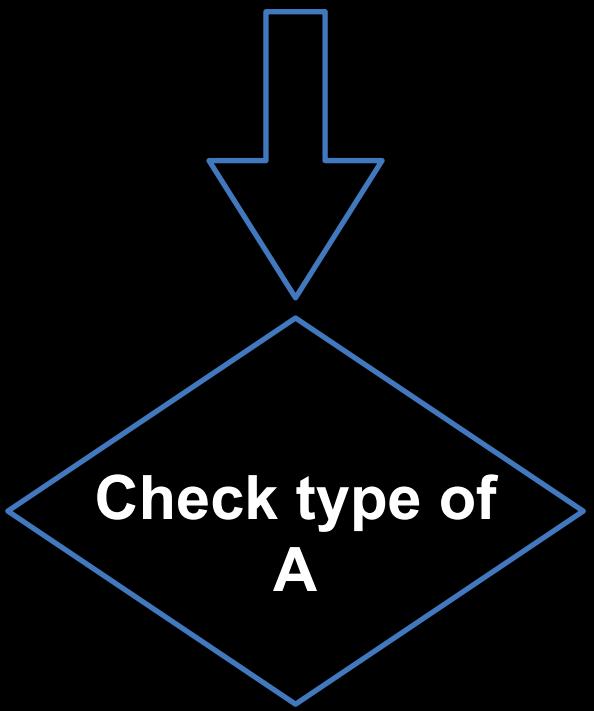
```
> swiftc main.swift  
> Error line 6: Cannot convert value of  
type 'String' into argument of type  
'Int'
```

Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```

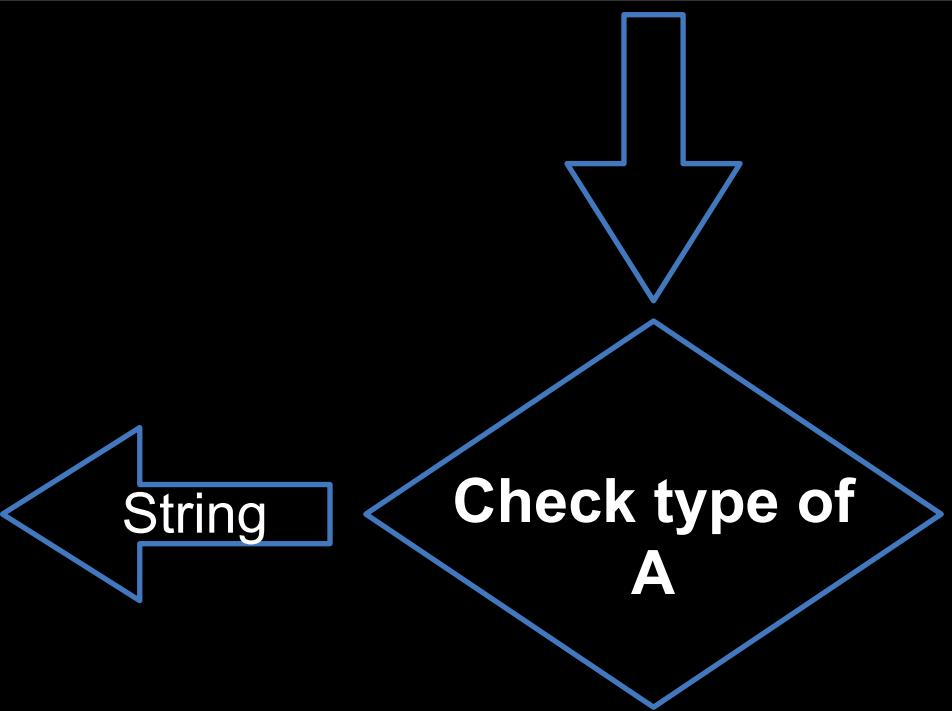
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



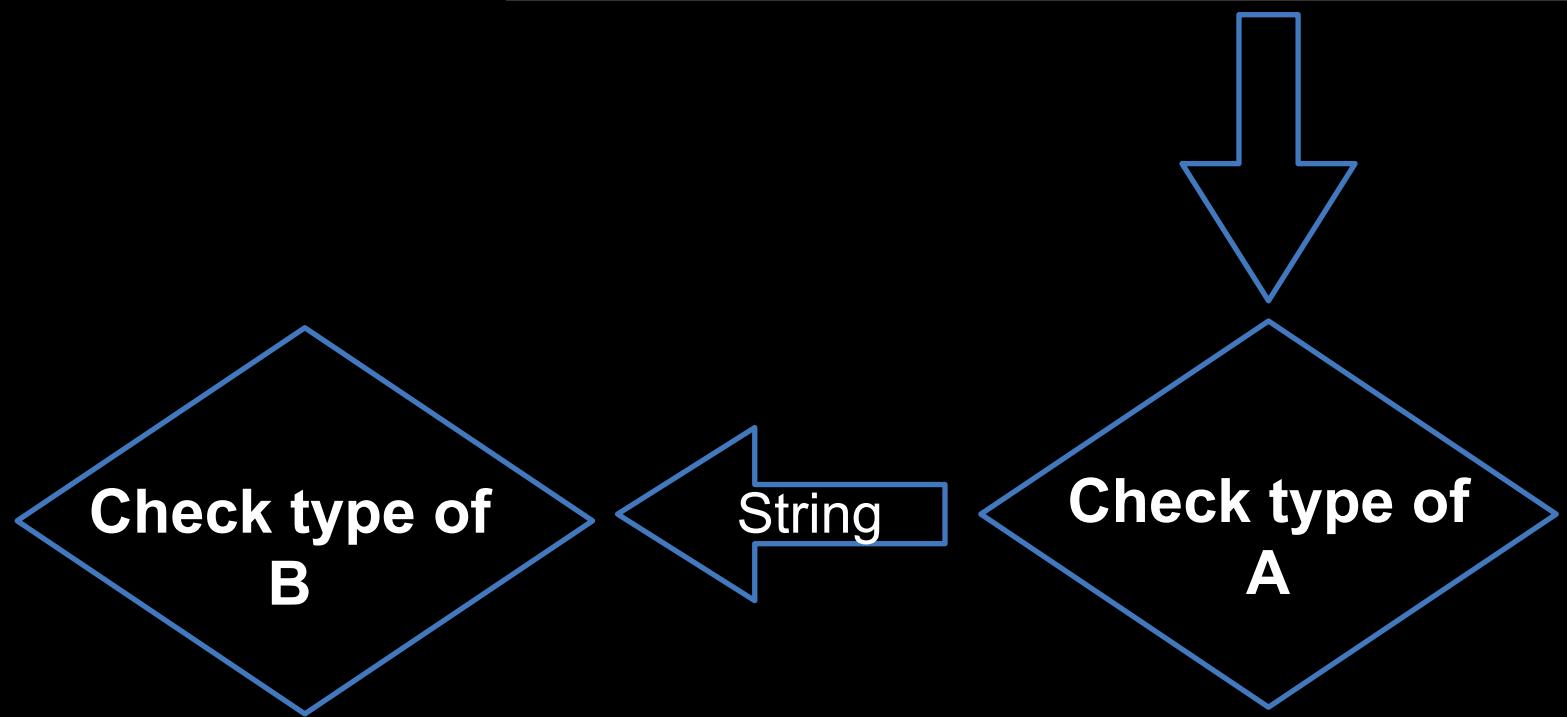
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



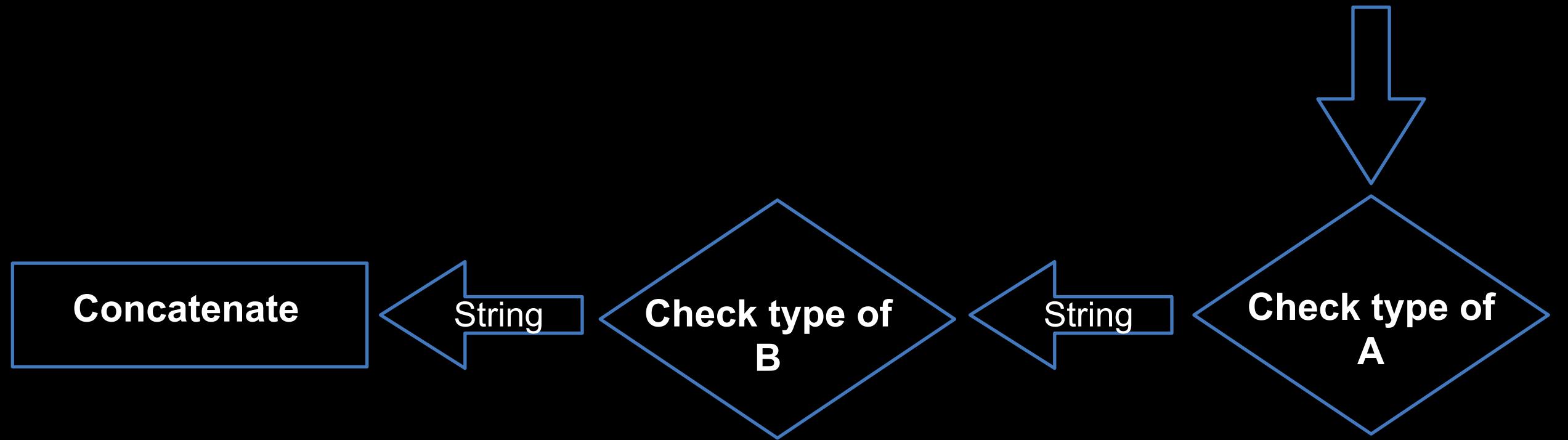
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



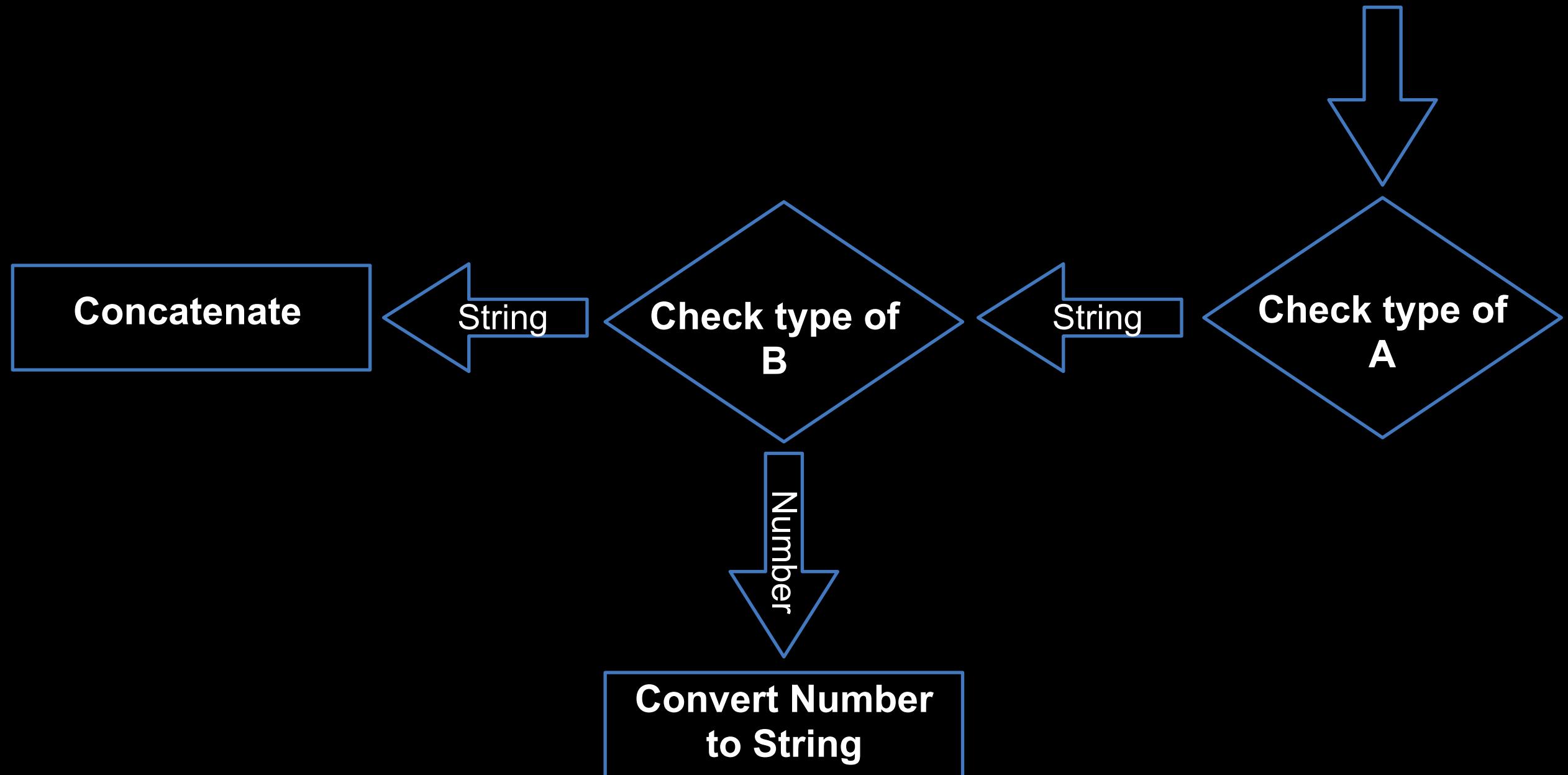
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



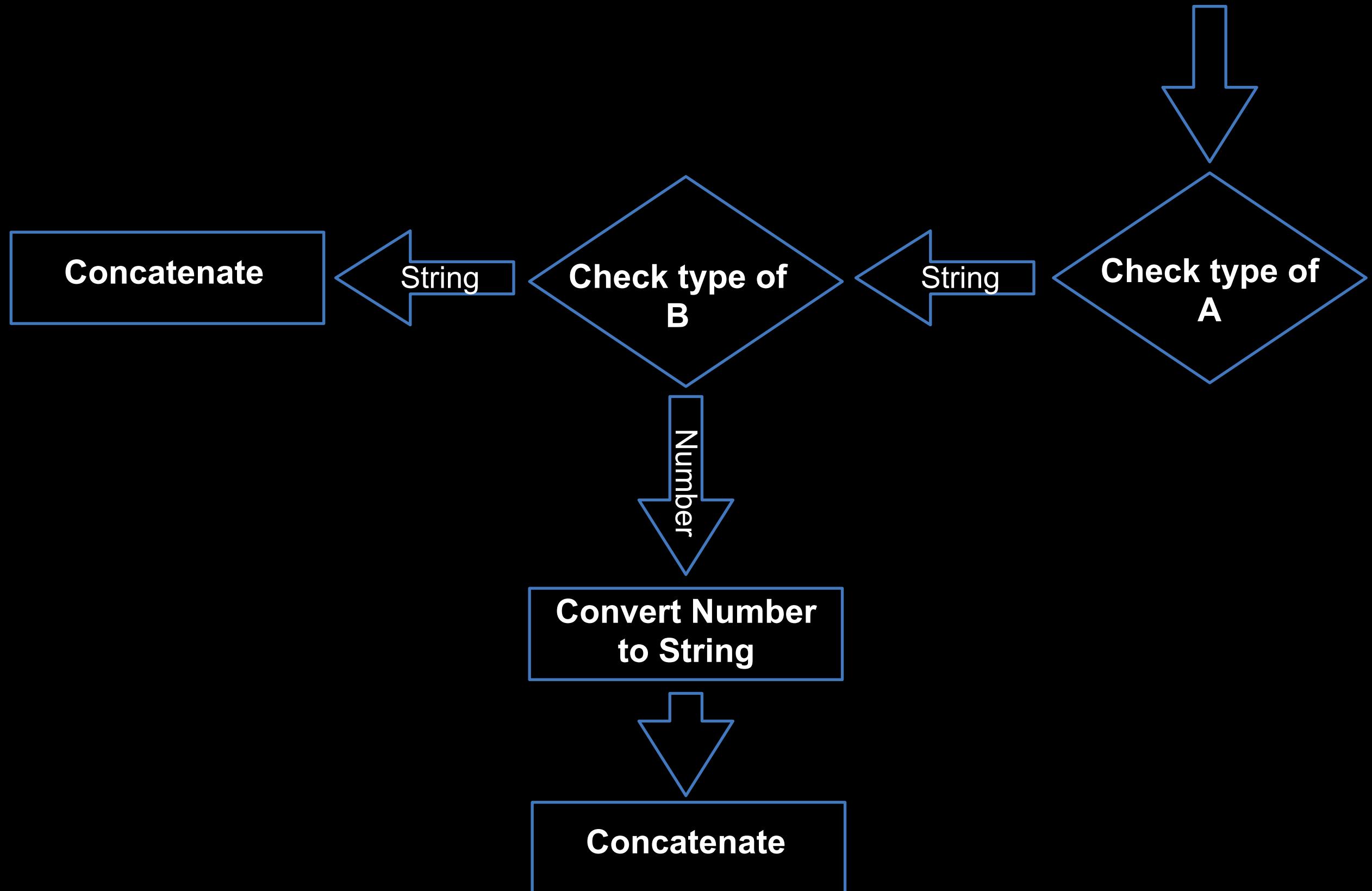
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



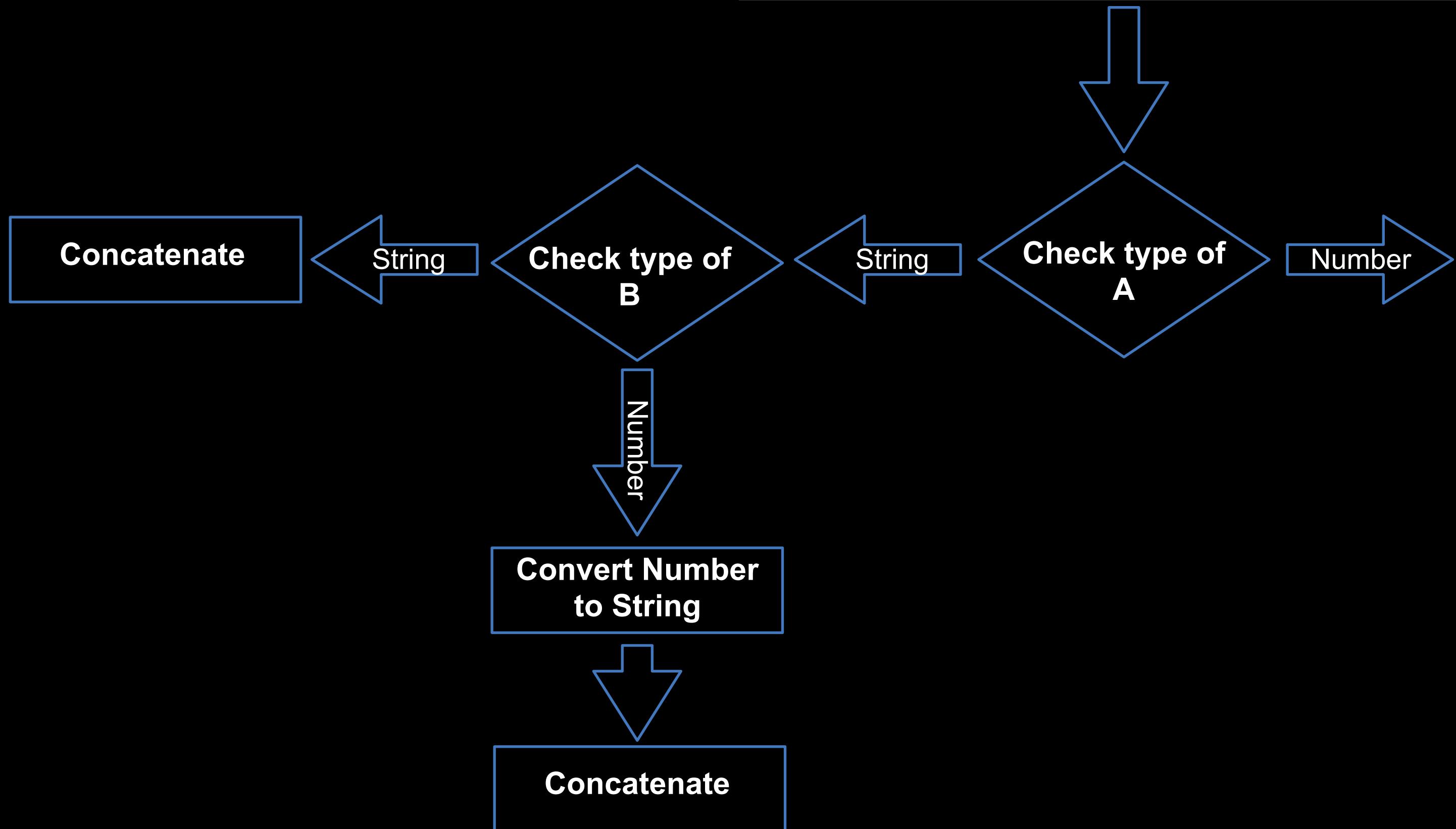
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



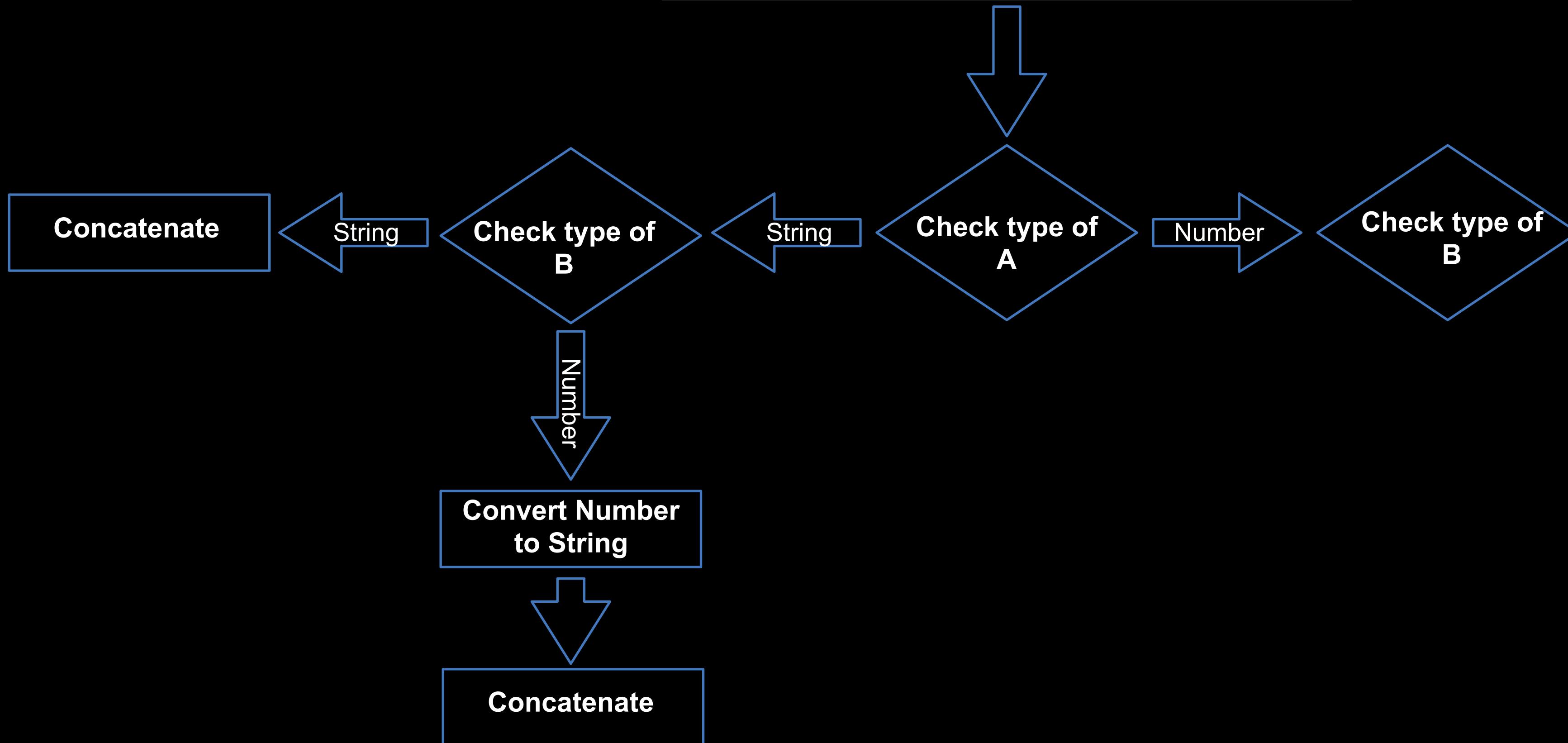
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



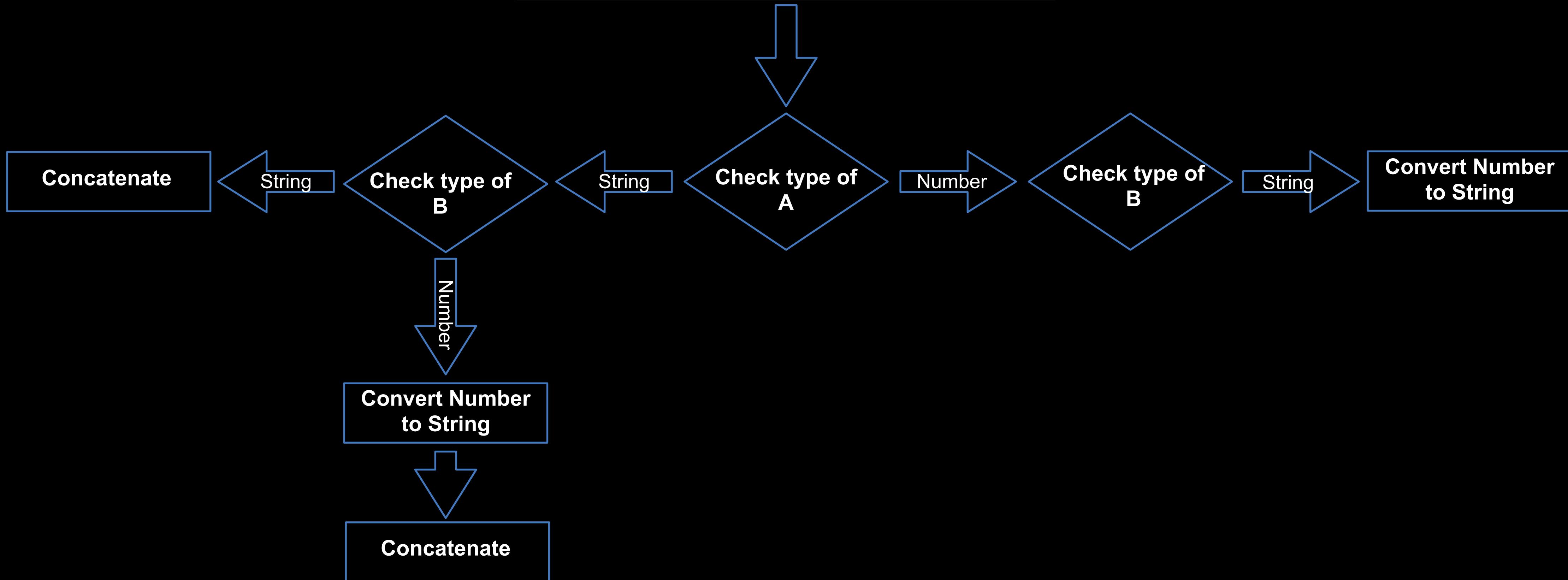
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



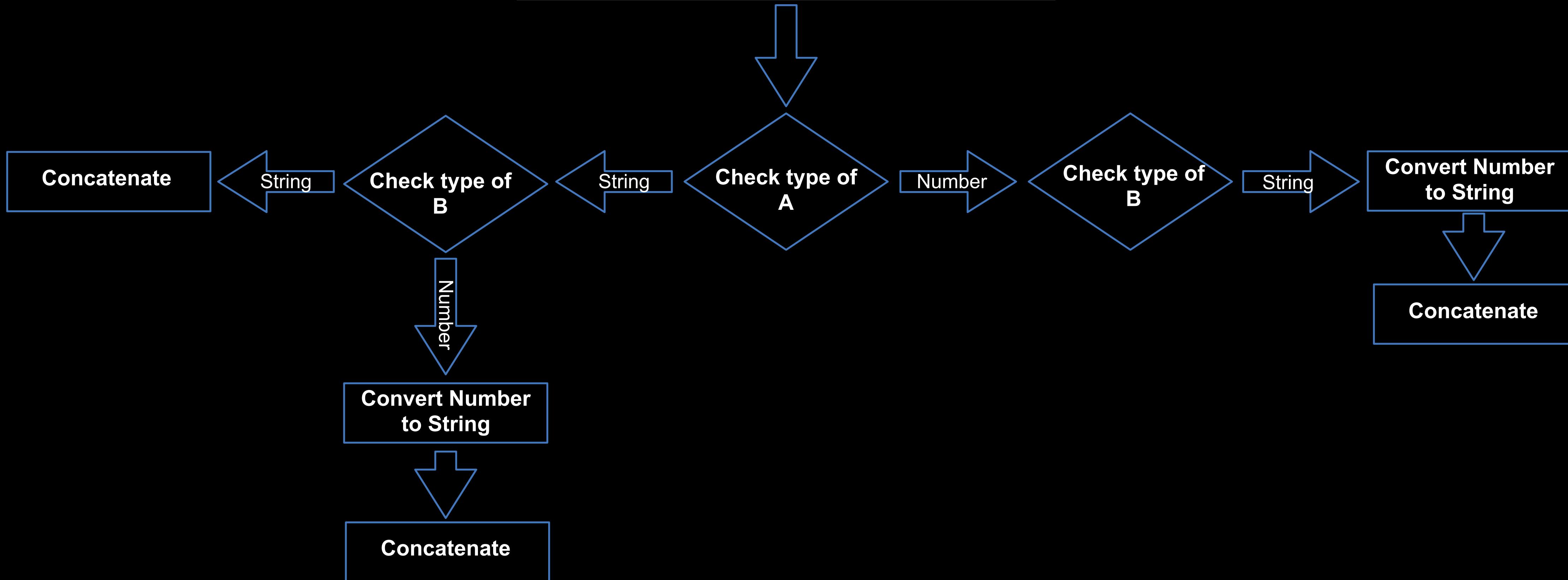
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



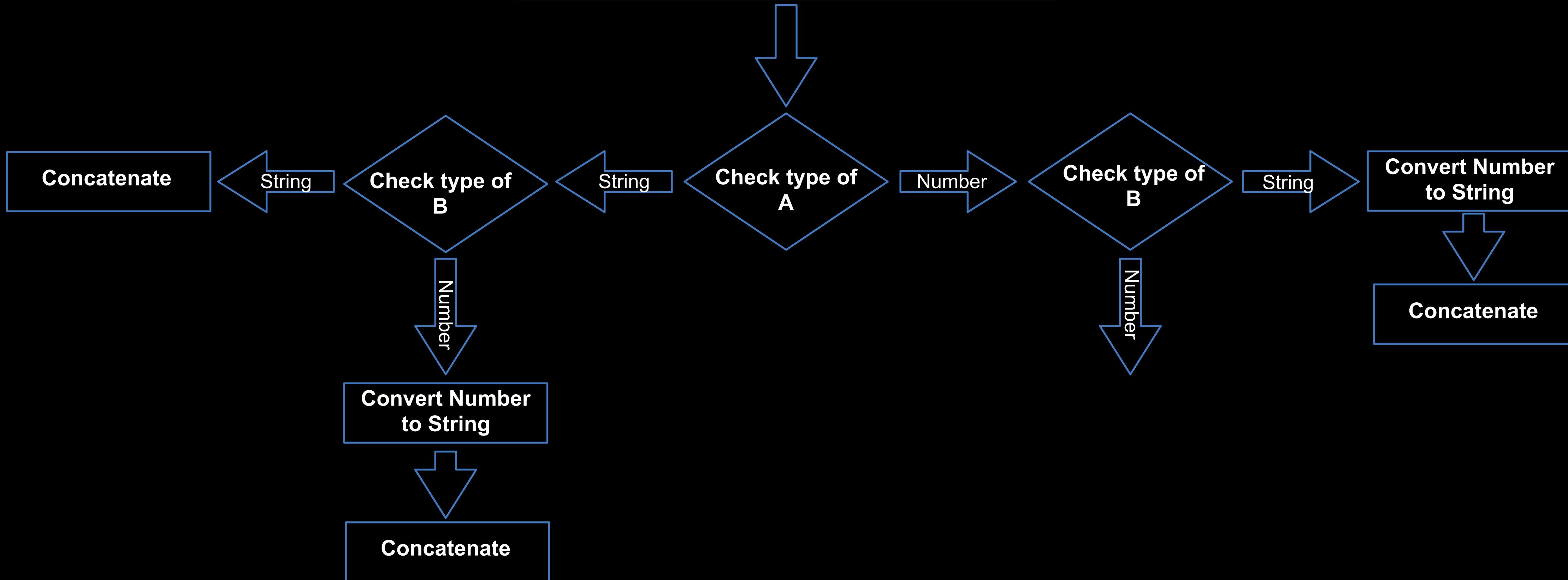
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



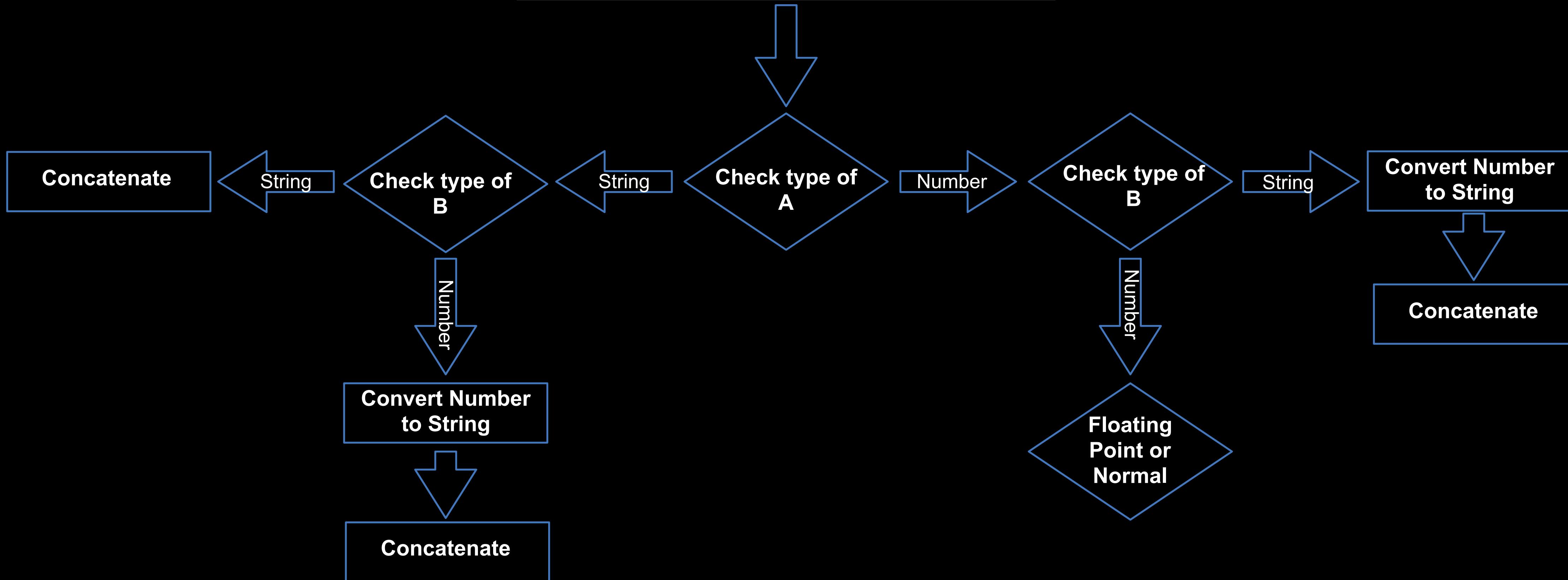
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



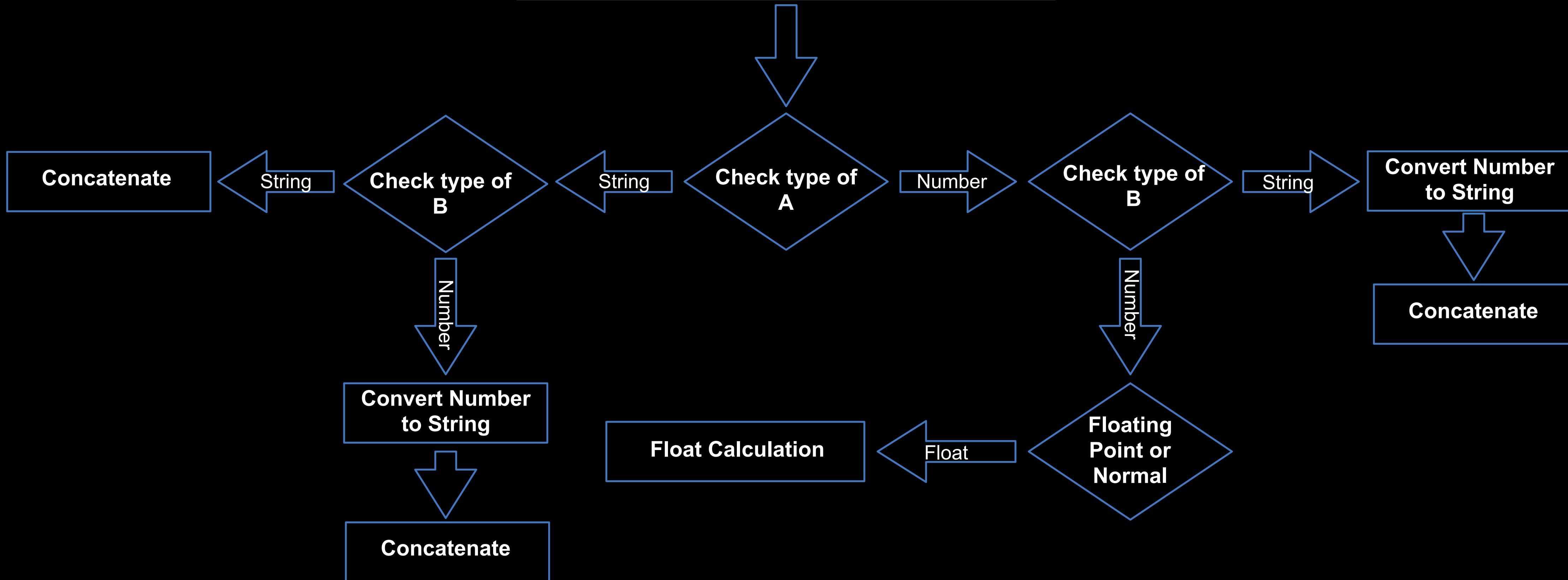
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



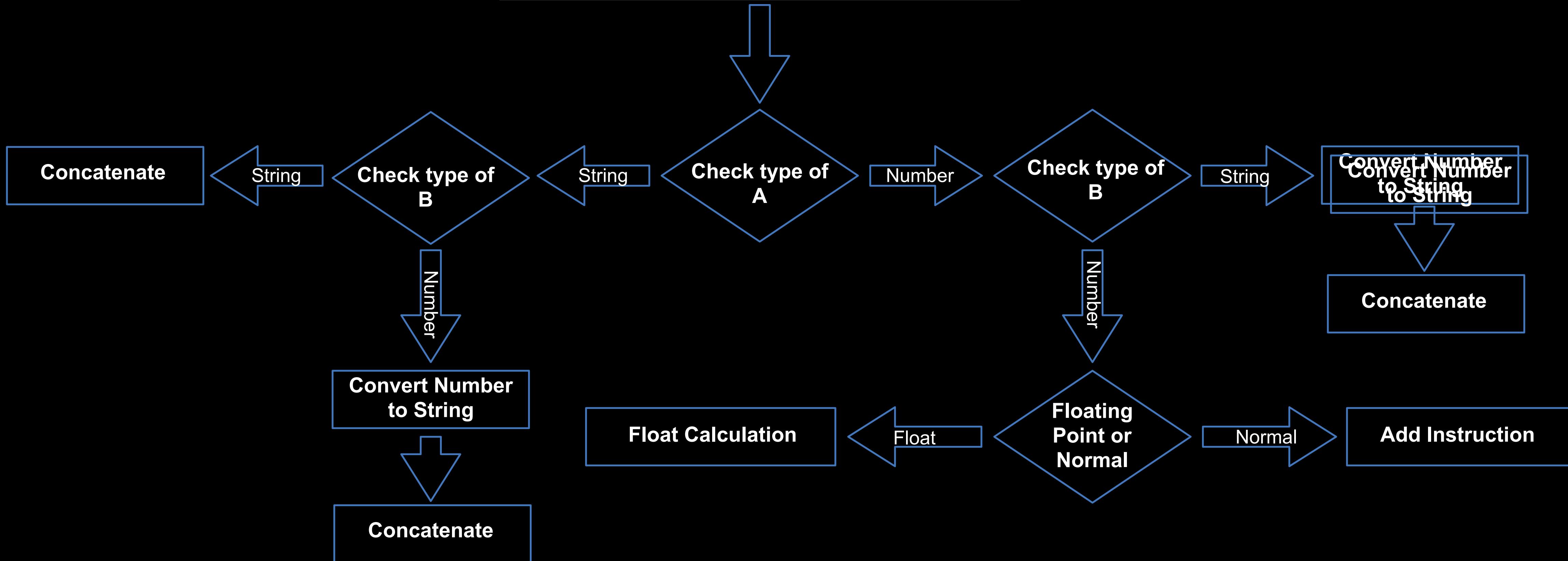
Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```



Runtime cost of dynamic typing

```
var add = function (a, b) {  
    return (a + b);  
}
```

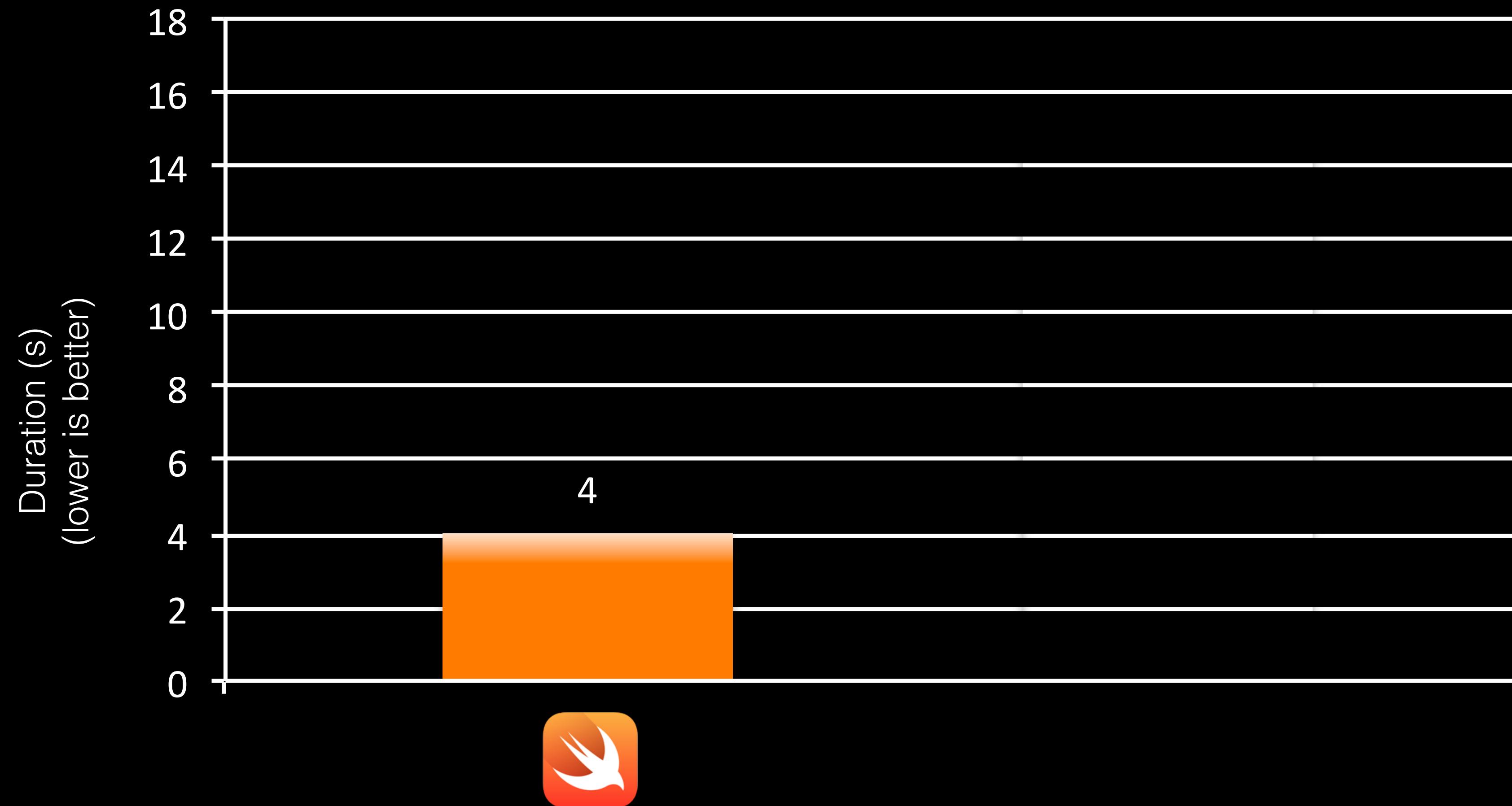


Typed vs Untyped Performance



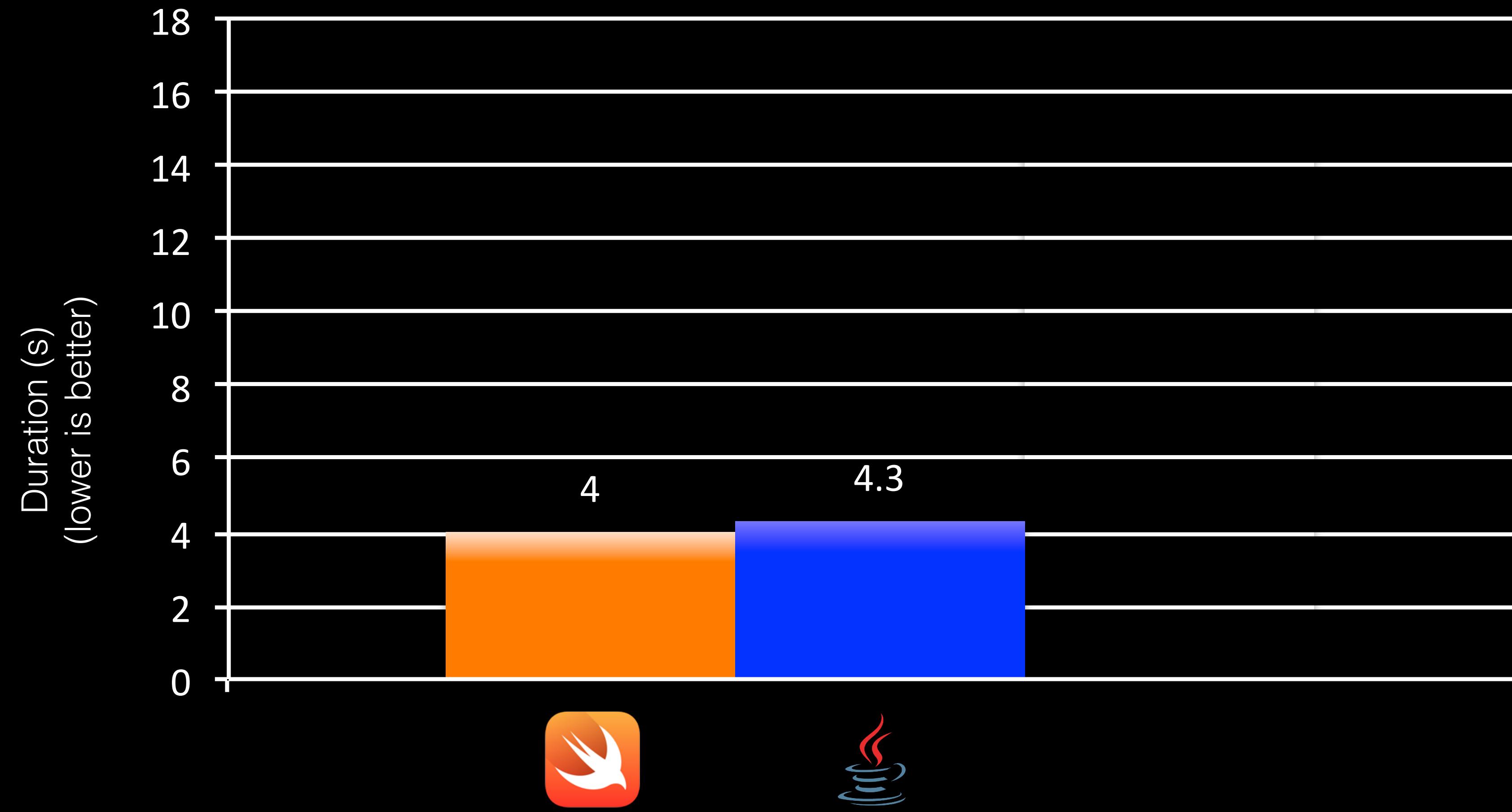
<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Typed vs Untyped Performance



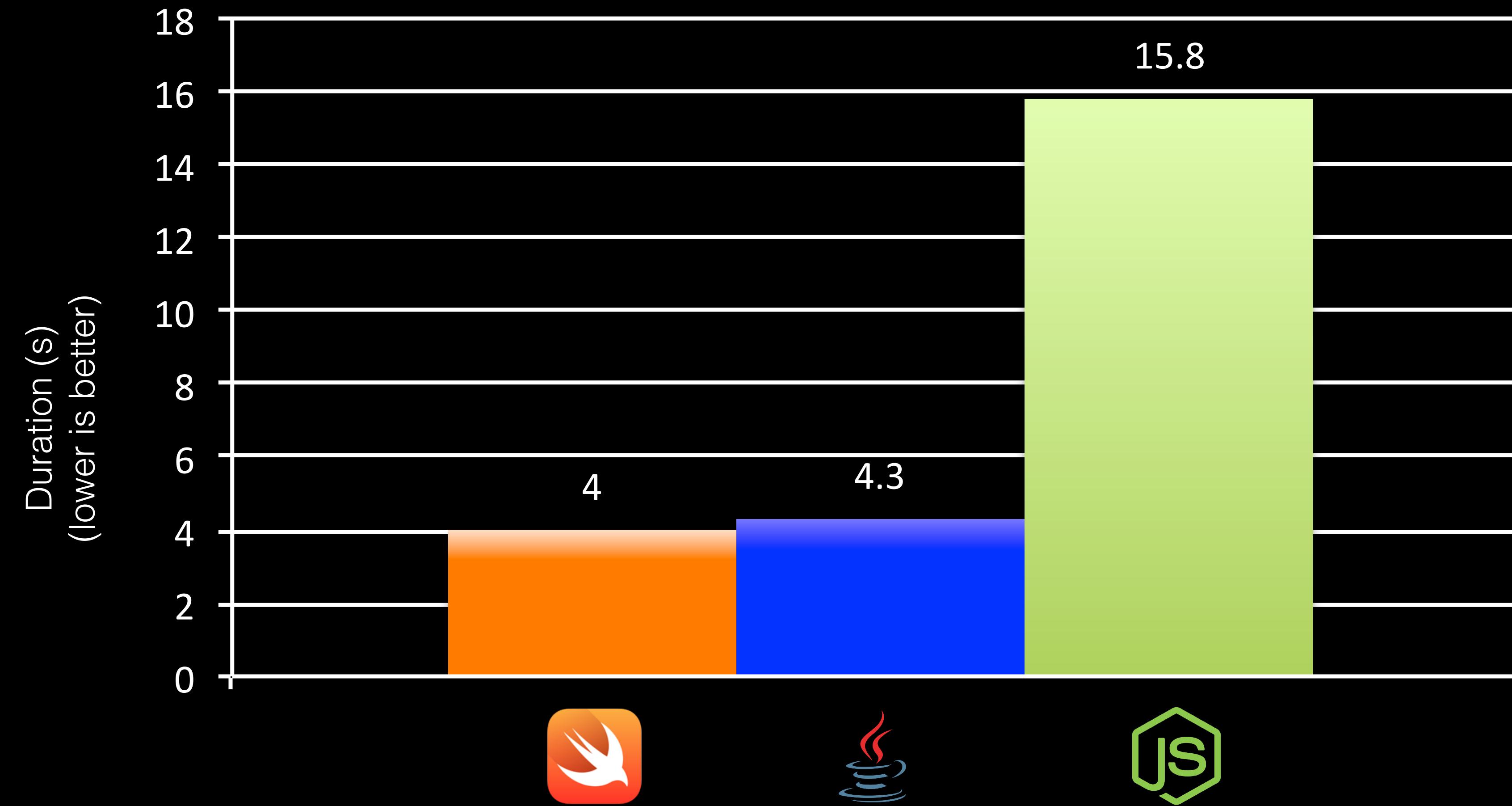
<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Typed vs Untyped Performance



<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

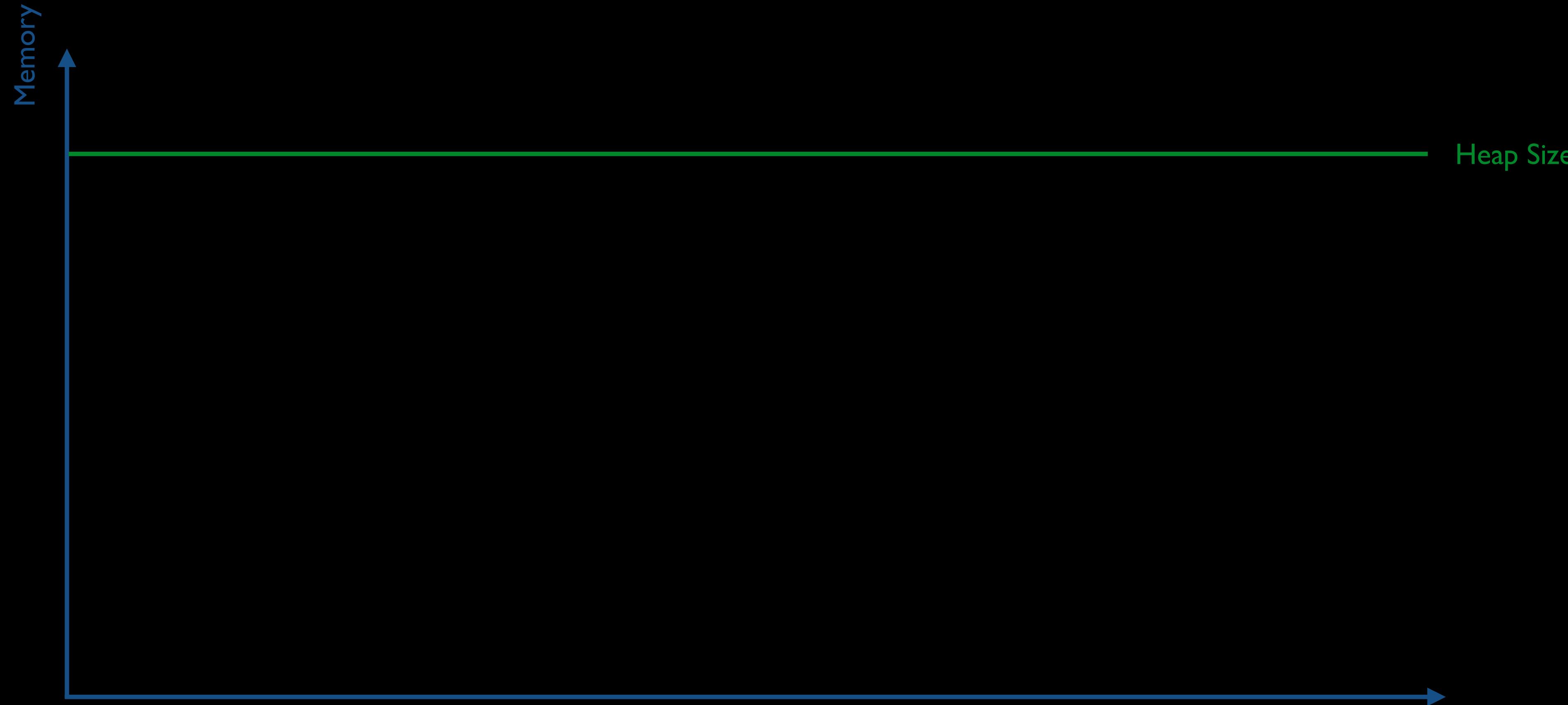
Typed vs Untyped Performance



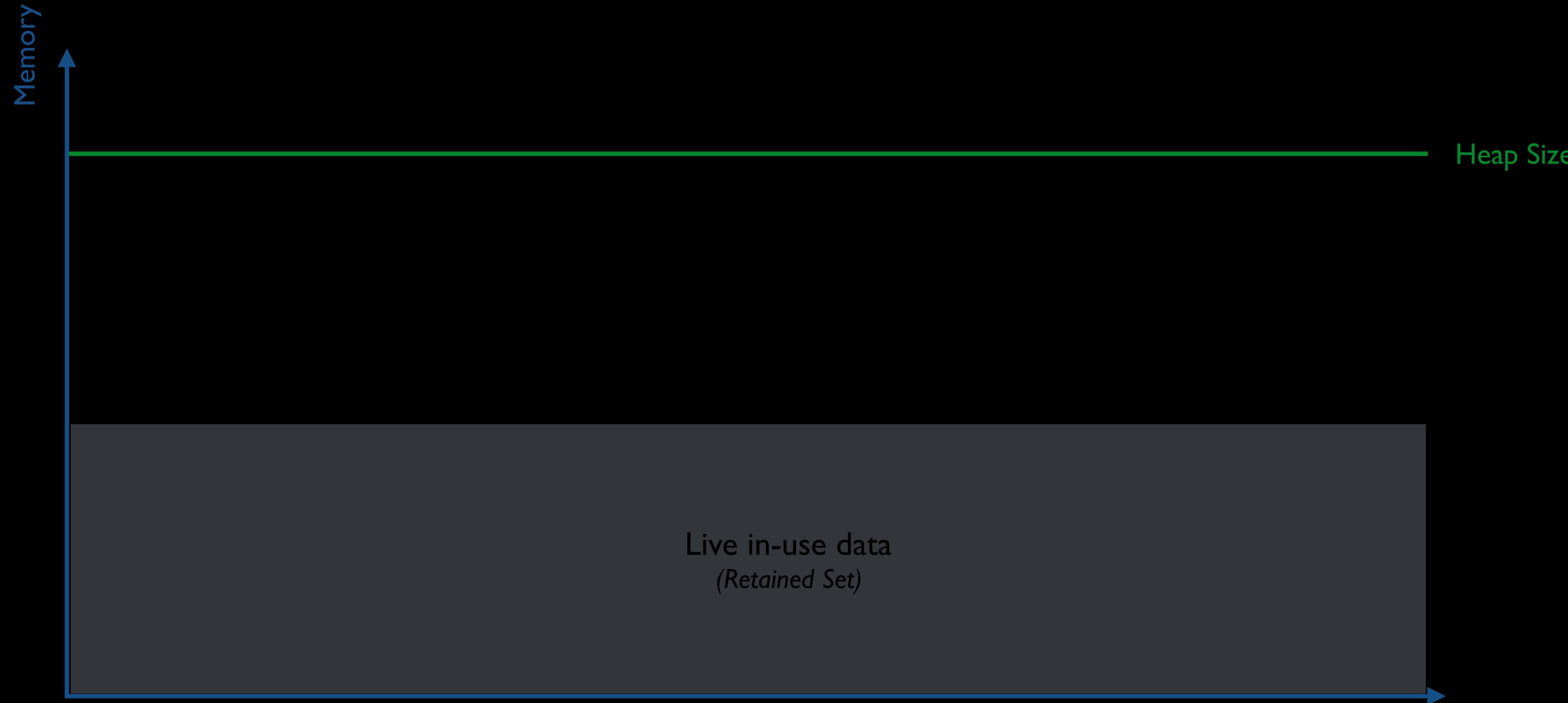
<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Memory Management

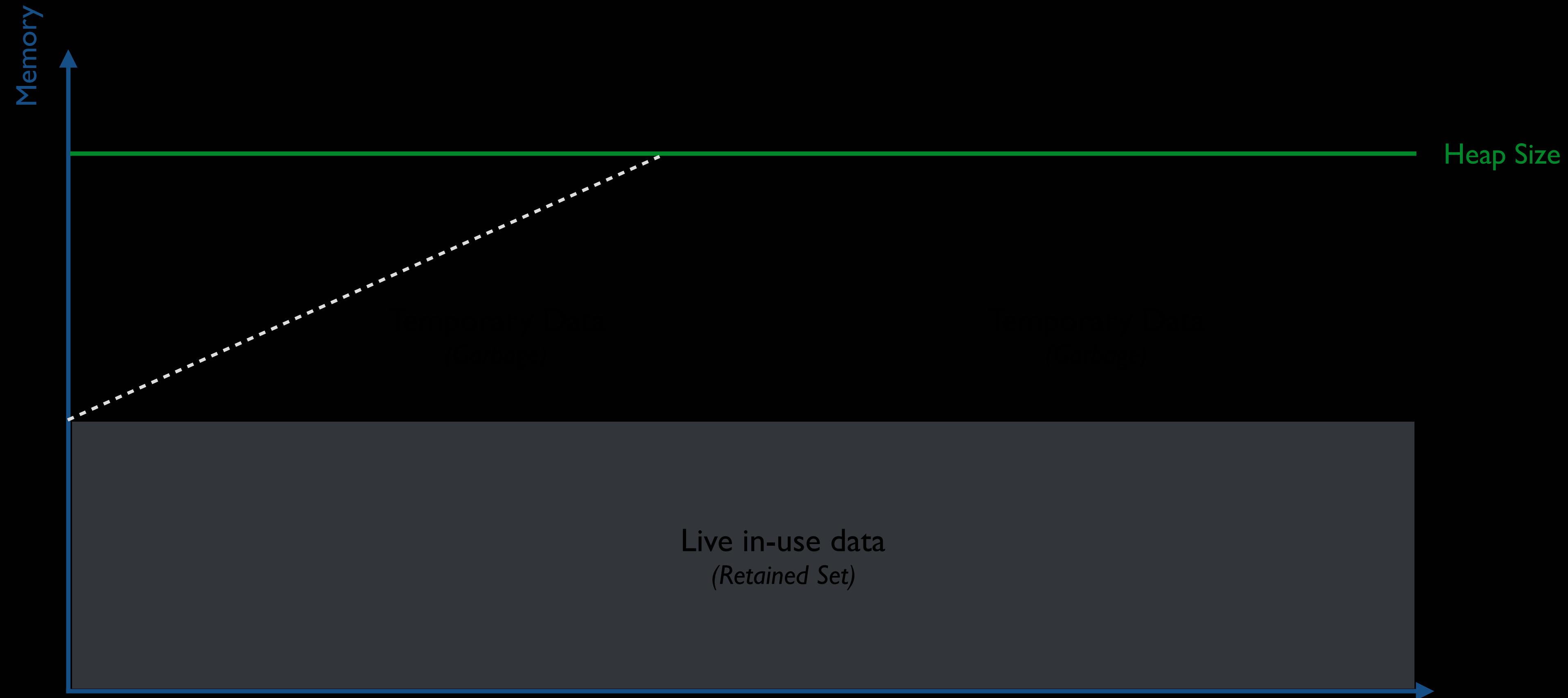
Garbage Collection Tradeoff



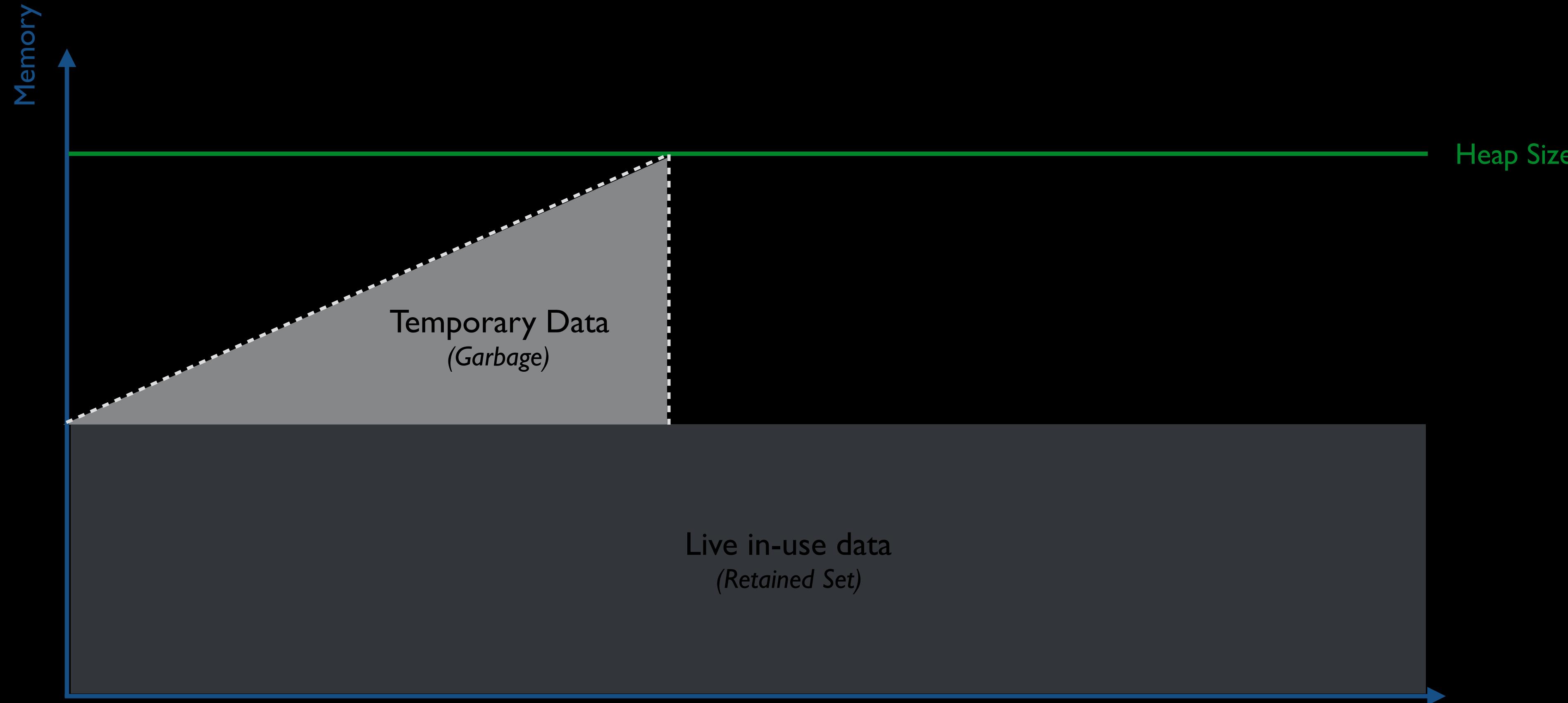
Garbage Collection Tradeoff



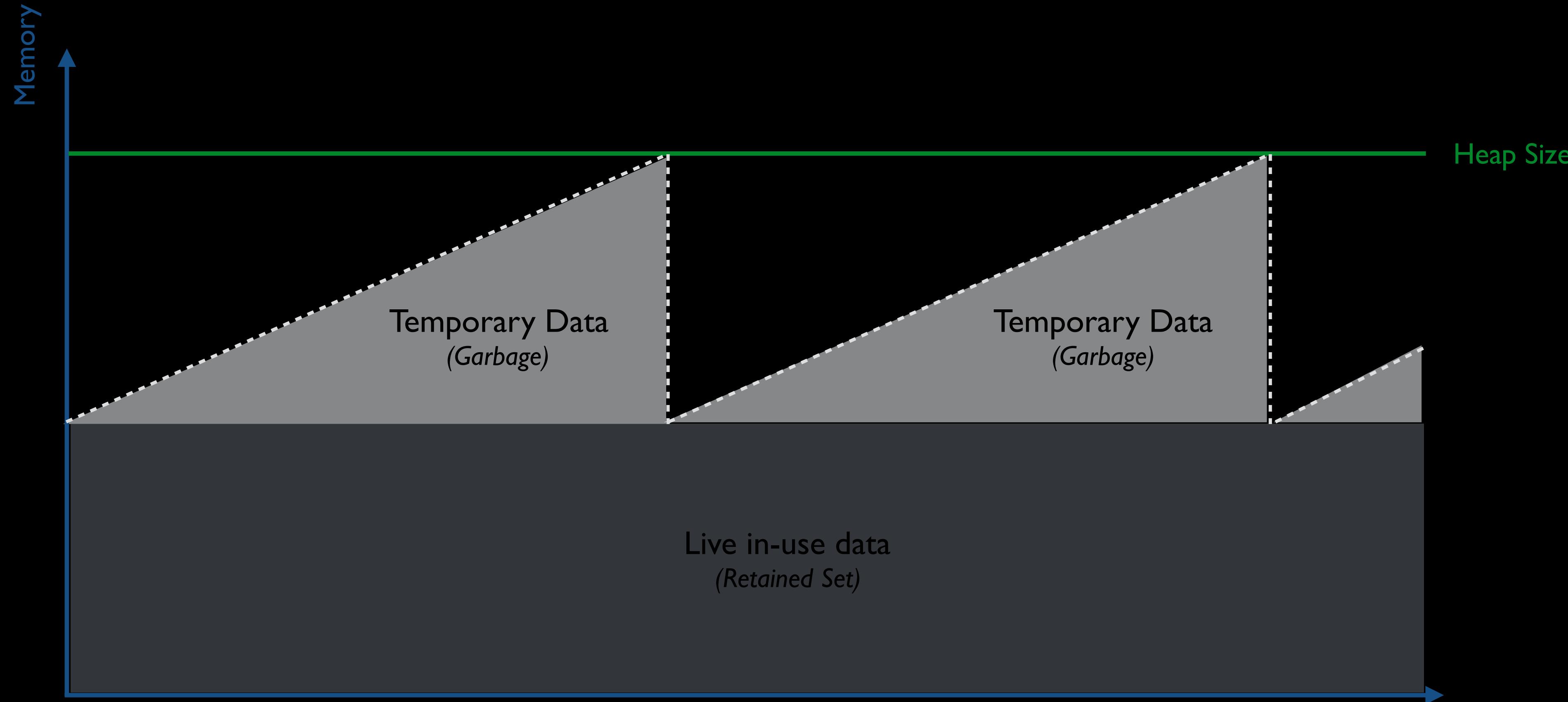
Garbage Collection Tradeoff



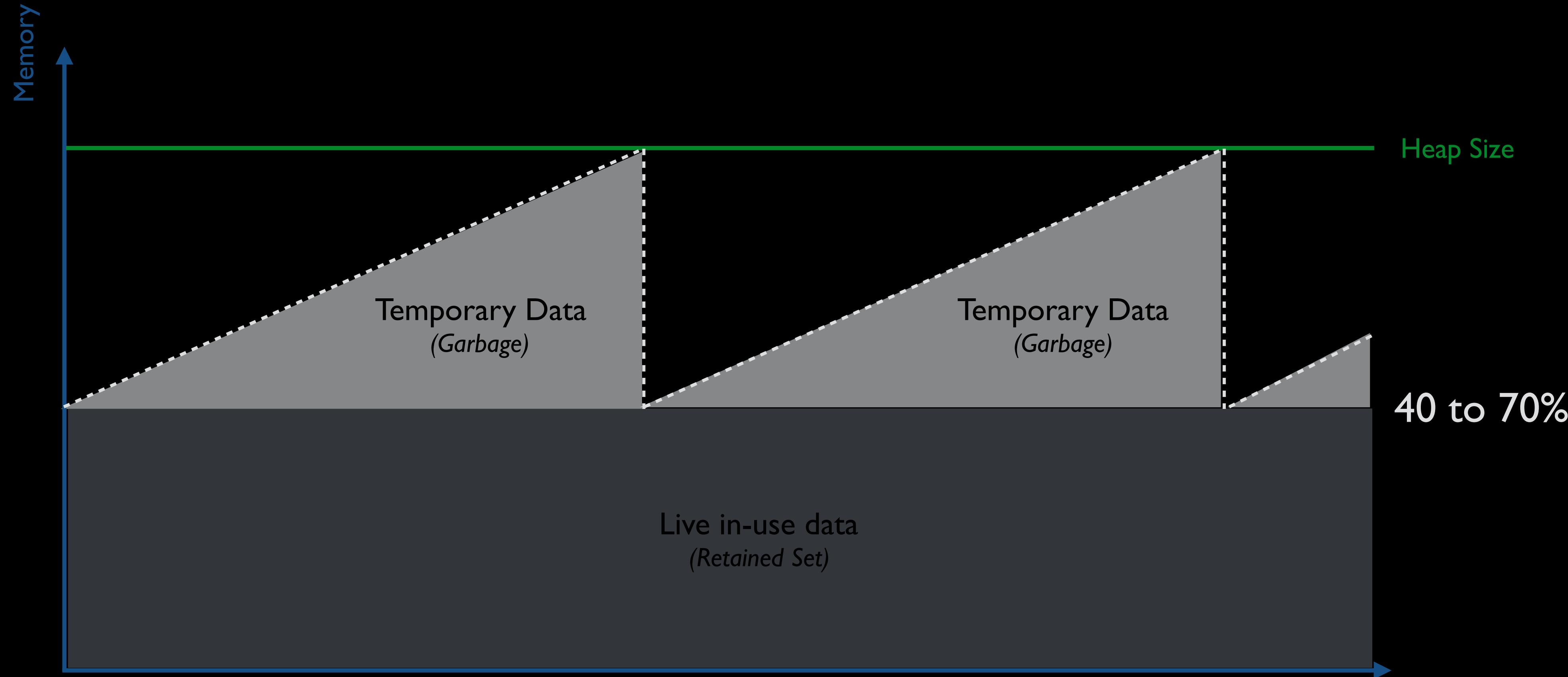
Garbage Collection Tradeoff



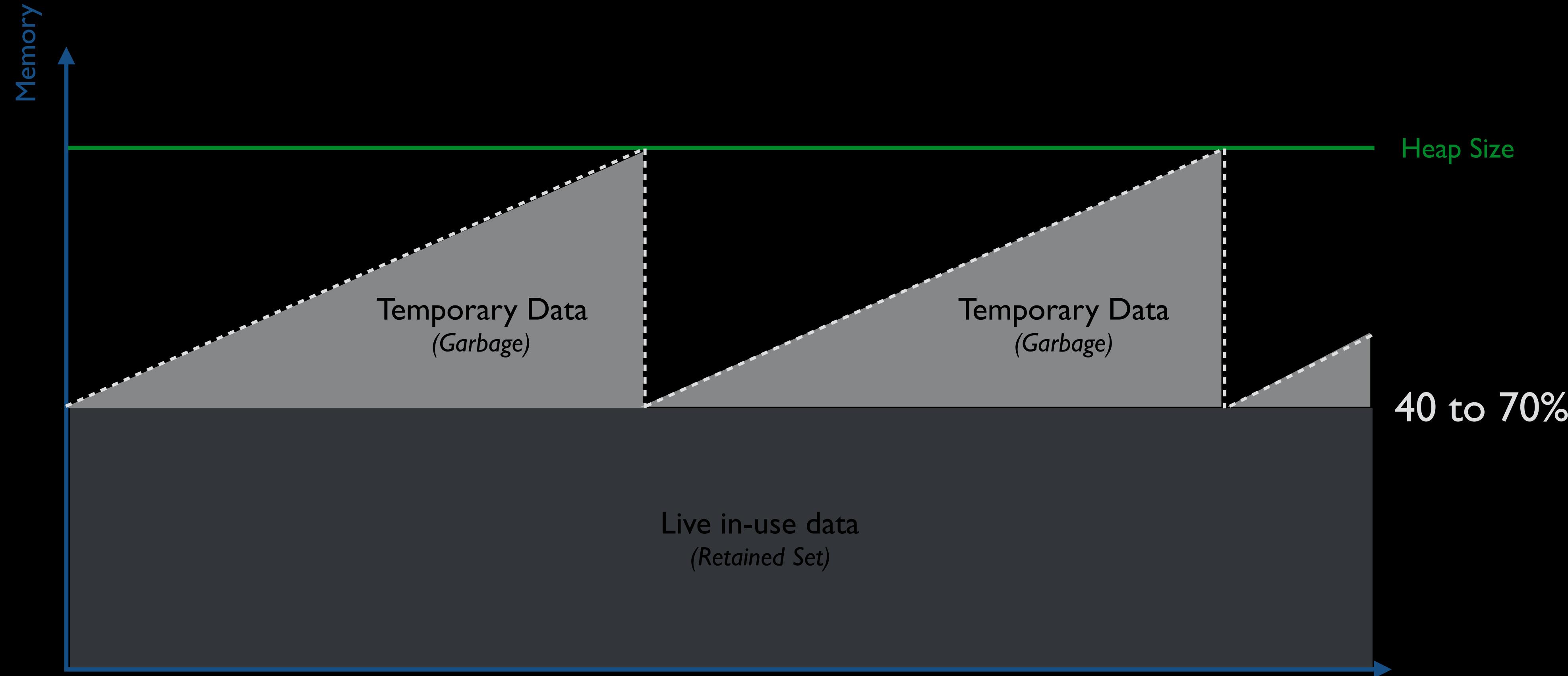
Garbage Collection Tradeoff



Garbage Collection Tradeoff

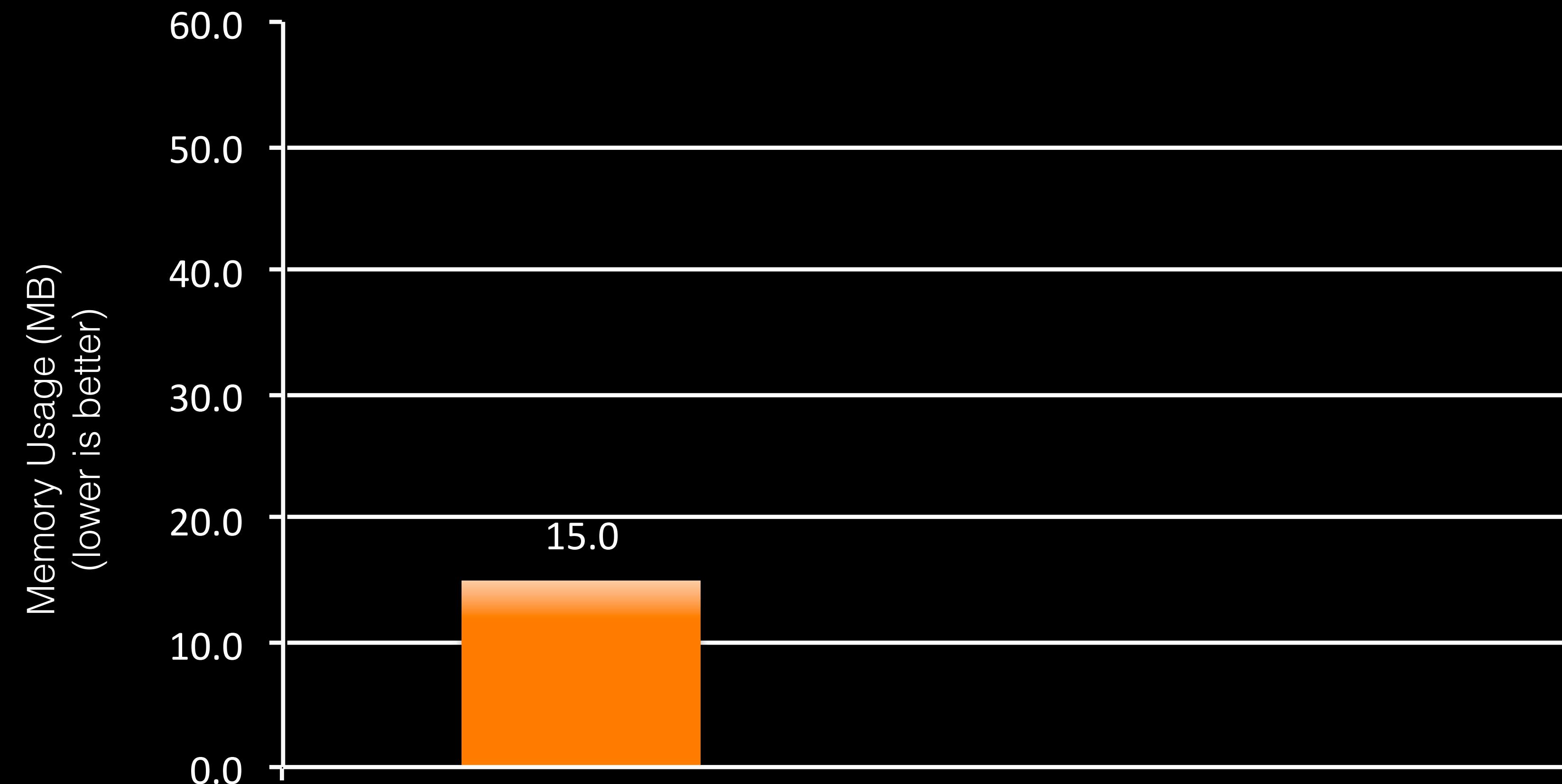


Garbage Collection Tradeoff



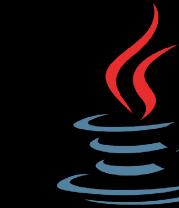
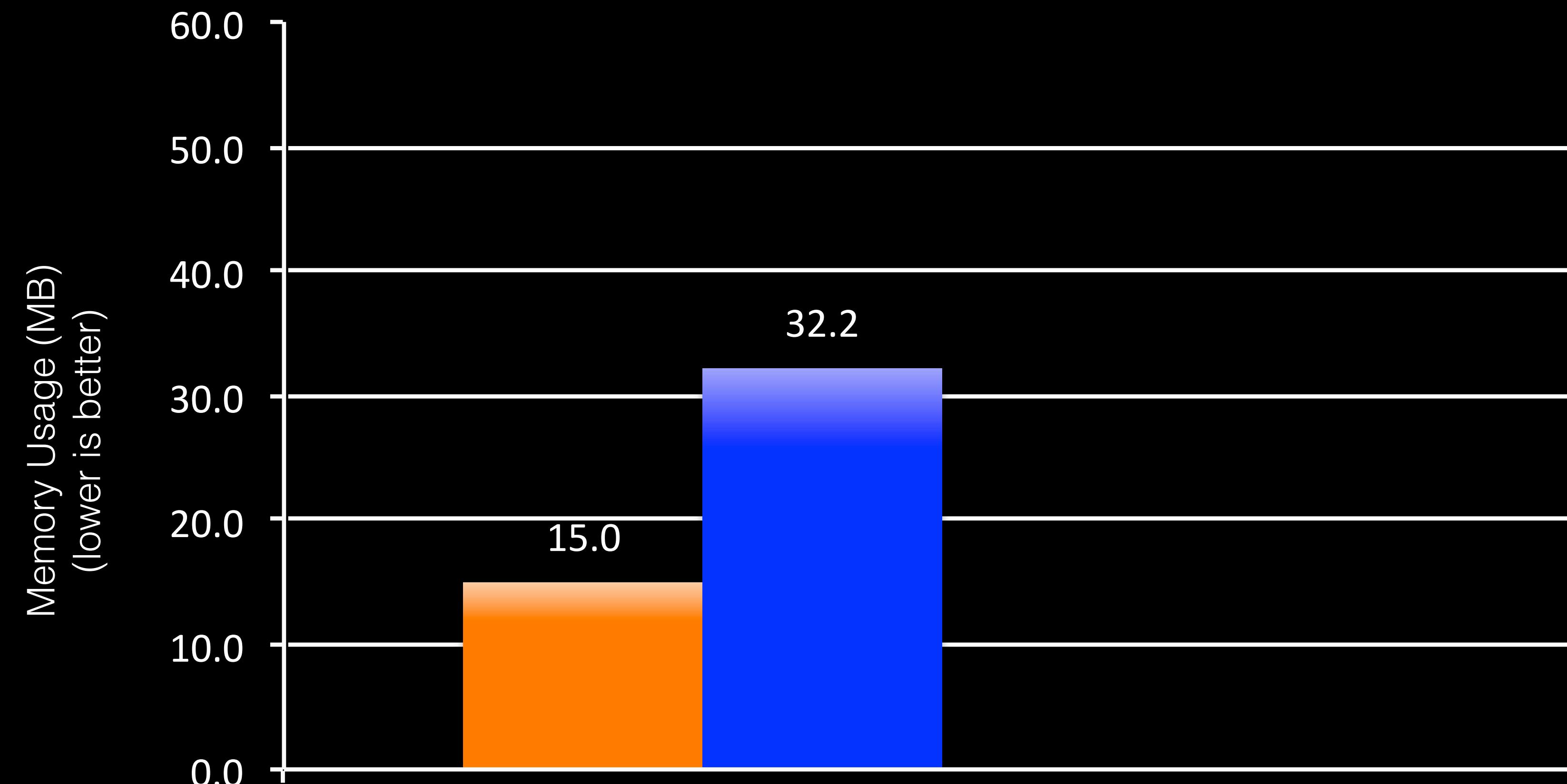
30 to 60% additional memory for additional performance

ARC vs Garage Collection



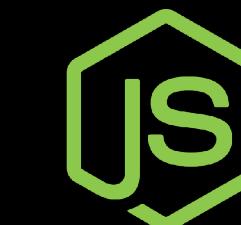
<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

ARC vs Garage Collection



<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

ARC vs Garage Collection



<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Cloud Costs

Provider	Type	Memory	CPUs	Cost/month
Amazon EC2	Linux t2.nano	512MB	1 vCPU	\$4.68
	Linux t2.micro	1GB	1 vCPU	\$9.36
	Linux t2.small	2GB	1 vCPU	\$18.72
	Linux t2.medium	4GB	2 vCPUs	\$37.44
Digital Ocean	Standard	512MB	1 Core Processor	\$5.00
	Standard	1GB	1 Core Processor	\$10.00
	Standard	2GB	2 Core Processors	\$20.00
Pivotal Cloud Foundry	App Instance	512MB	4 vCPUs	\$10.80
	App Instance	1GB	4 vCPUs	\$21.60
	App Instance	2GB	4 vCPUs	\$43.20
Heroku	Standard 1x	512MB	1 Share	\$25.00
	Standard 2x	1GB	2 Shares	\$50.00
IBM Bluemix	Instant Runtimes	512MB	4 vCPUs	\$24.15
	Instant Runtimes	1GB	4 vCPUs	\$49.35
	Containers	512MB	4 vCPUs	\$10.22
	Containers	1GB	4 vCPUs	\$20.59

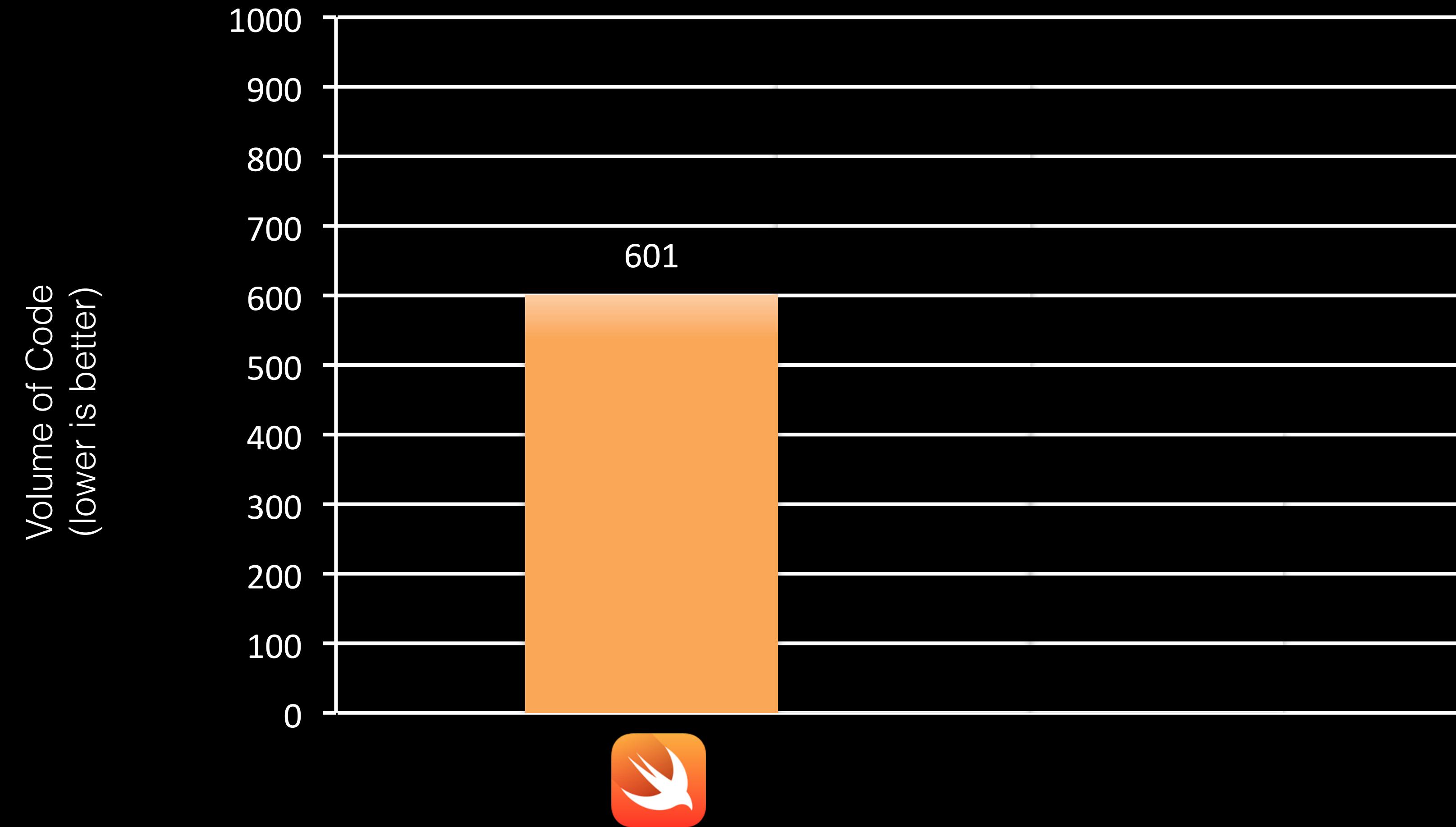
Developer Productivity

Typed vs Untyped Developer Productivity



<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Typed vs Untyped Performance



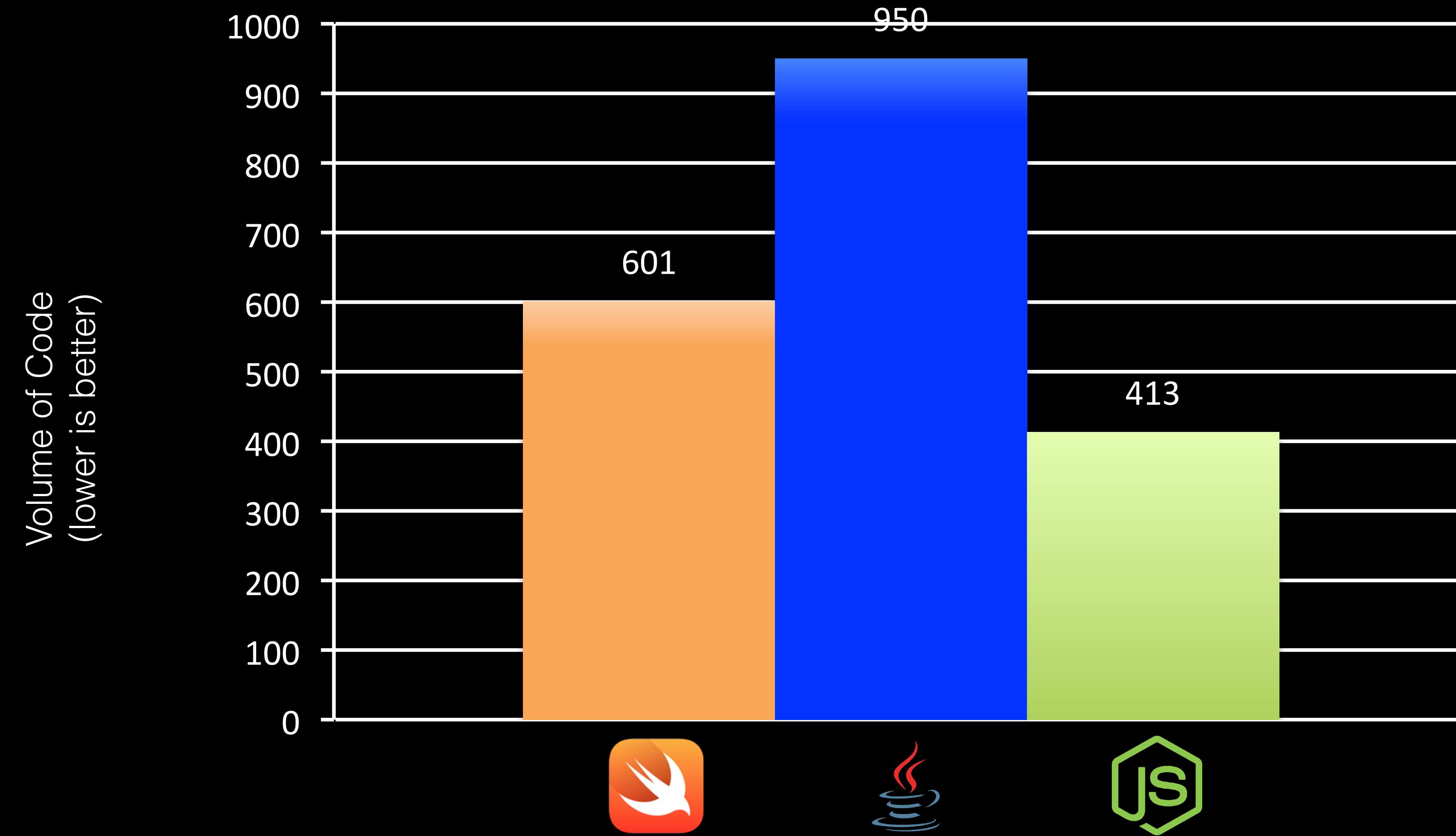
<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Typed vs Untyped Performance



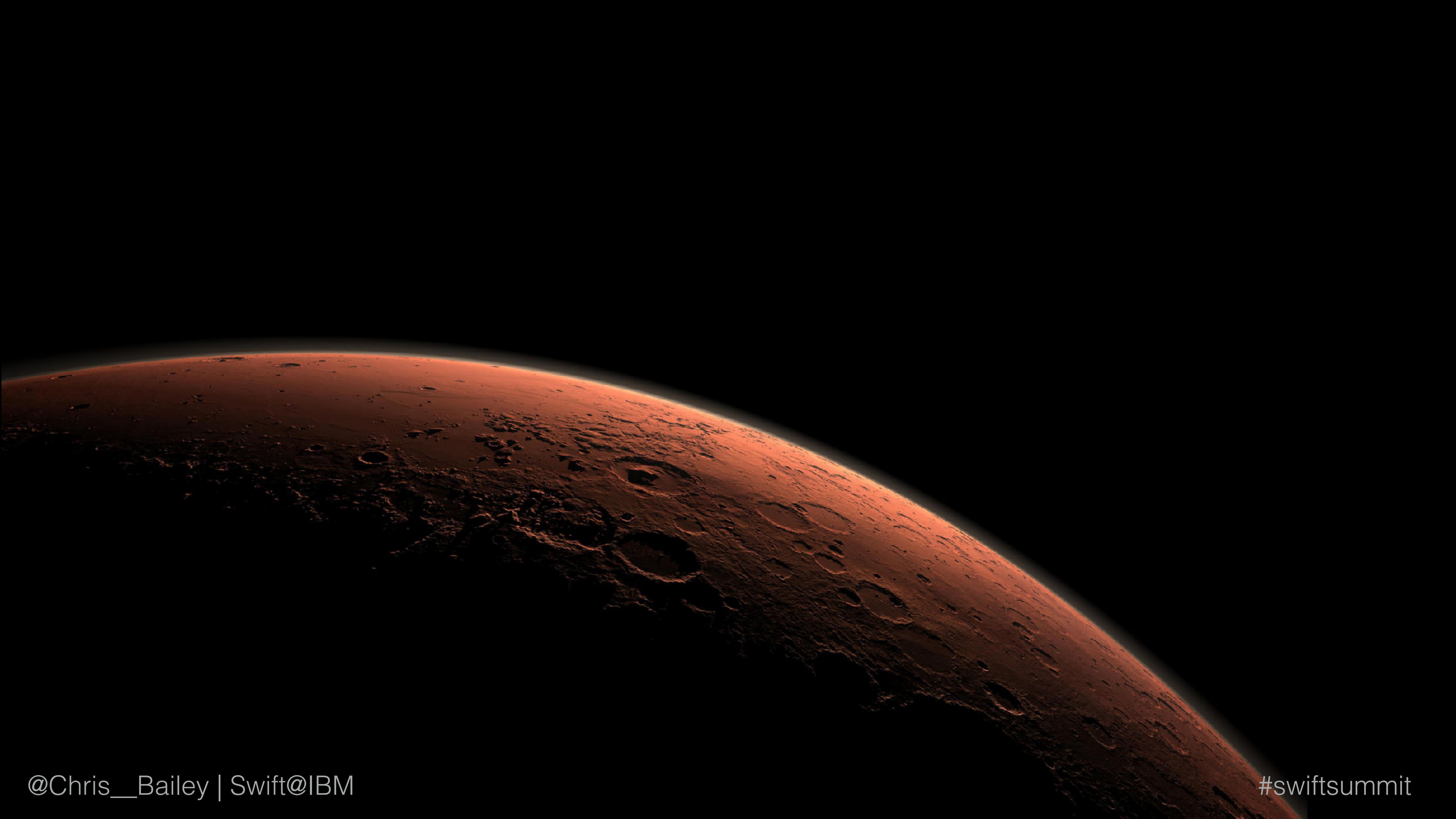
<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

Typed vs Untyped Performance



<http://benchmarksgame.alioth.debian.org/u64q/performance.php?test=spectralnorm>

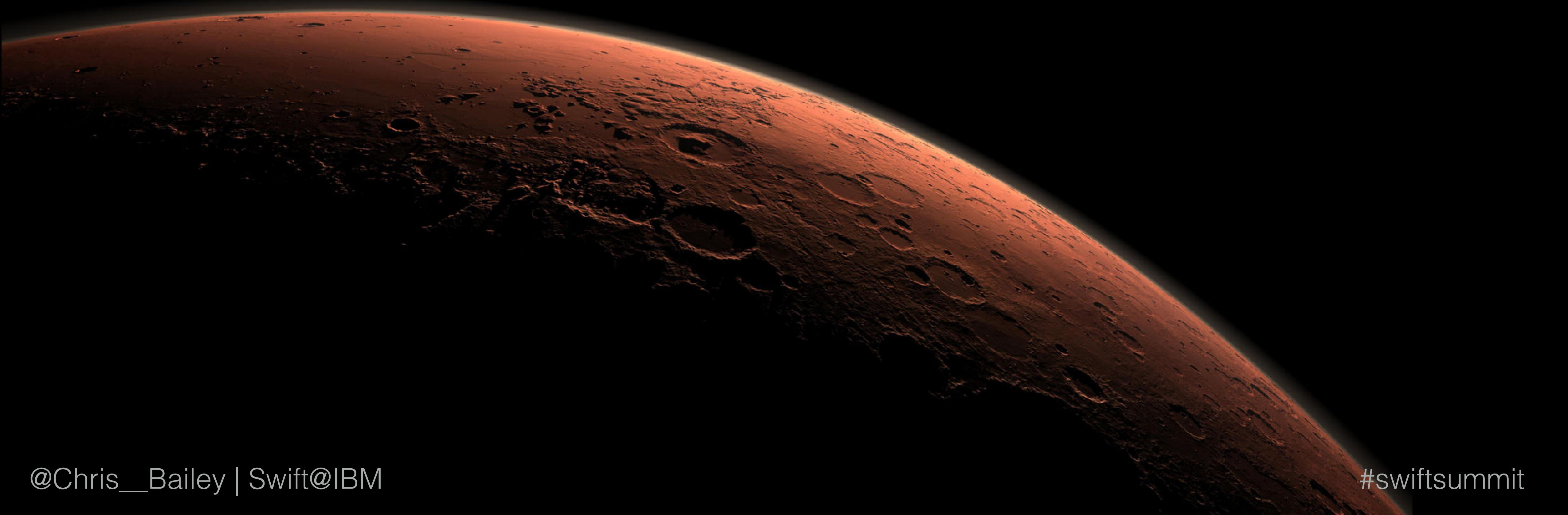
Fullstack Development



@Chris_Bailey | Swift@IBM

#swiftsummit

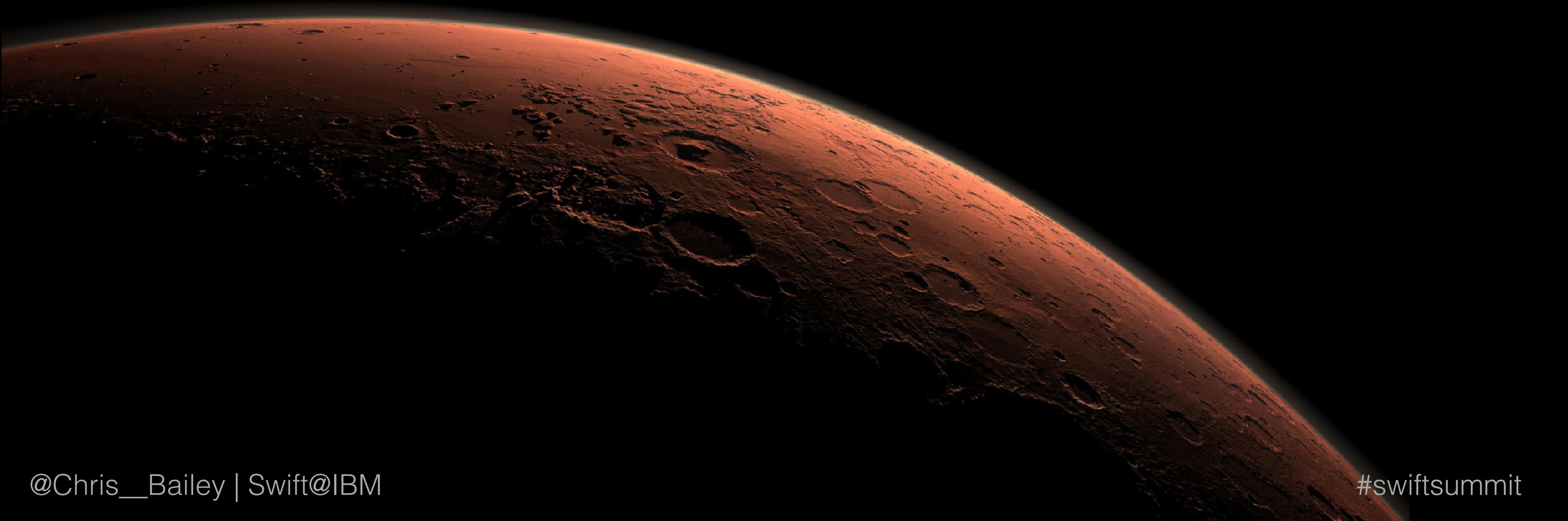
MARS



1998

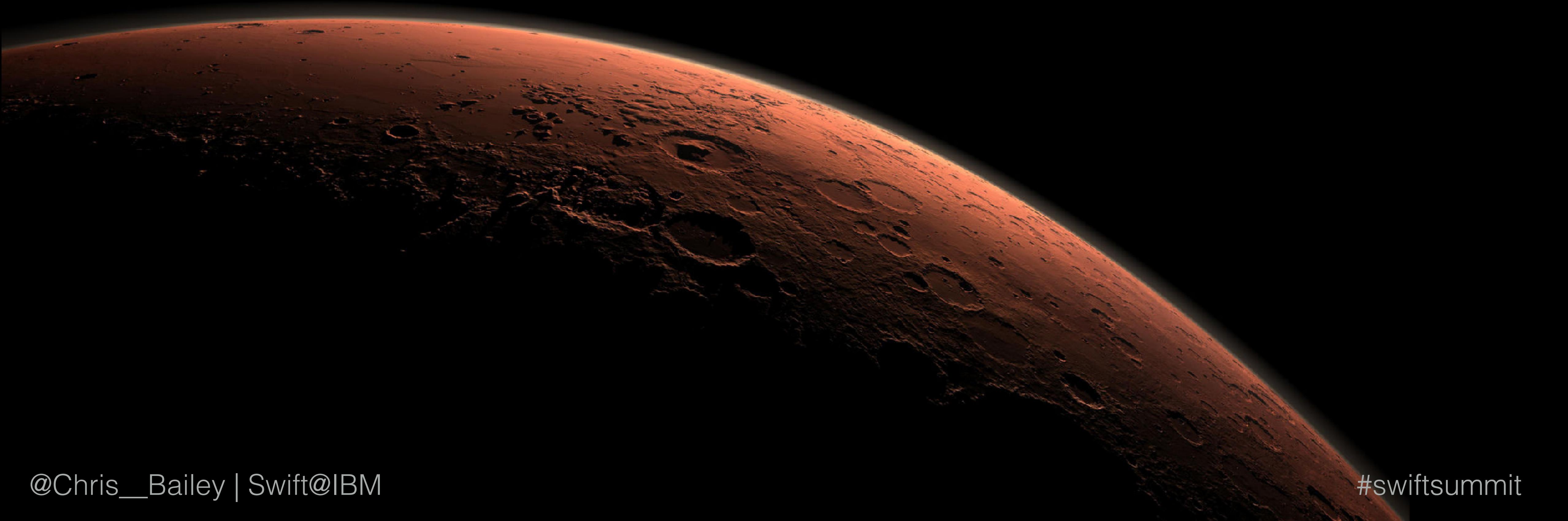
MARS CLIMATE ORBITER

2024



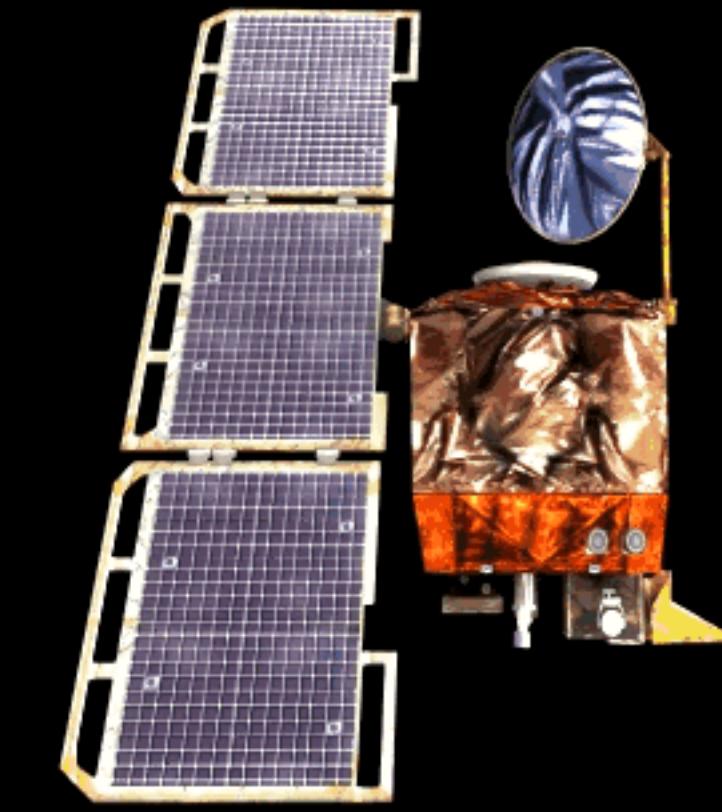
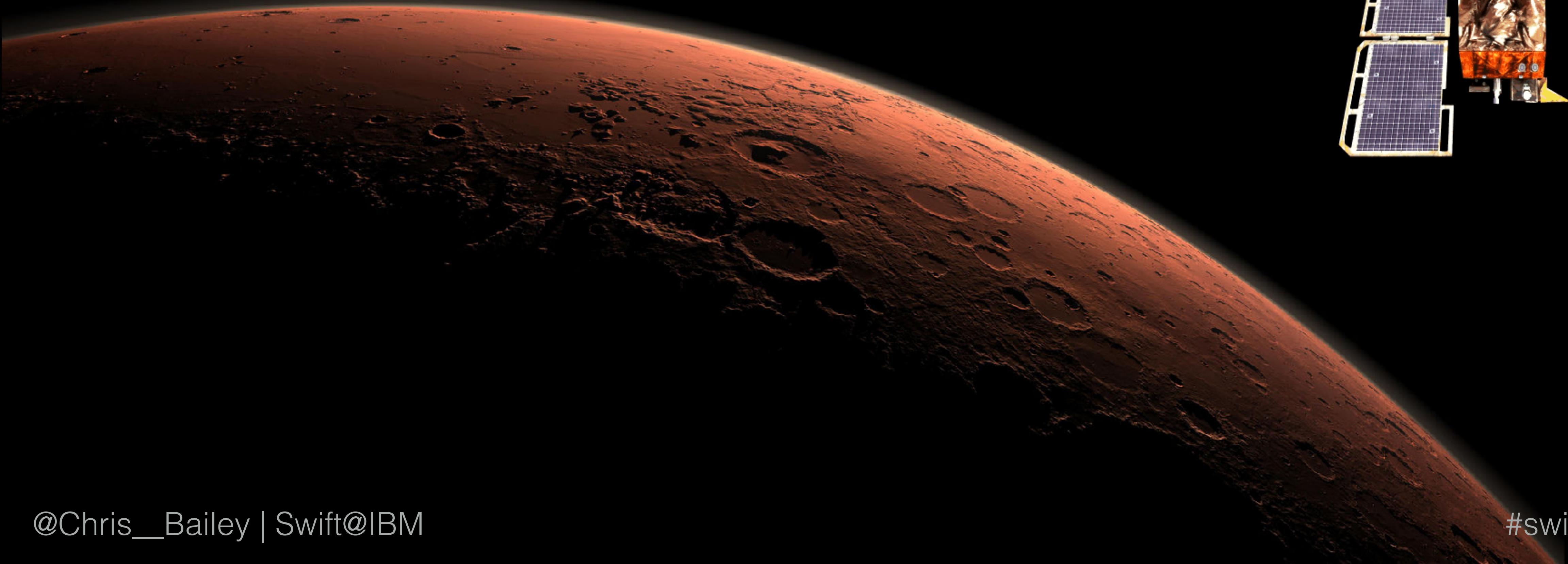
1998

MARS CLIMATE ORBITER
DEC 11: LAUNCH FROM CAPE CANAVERAL



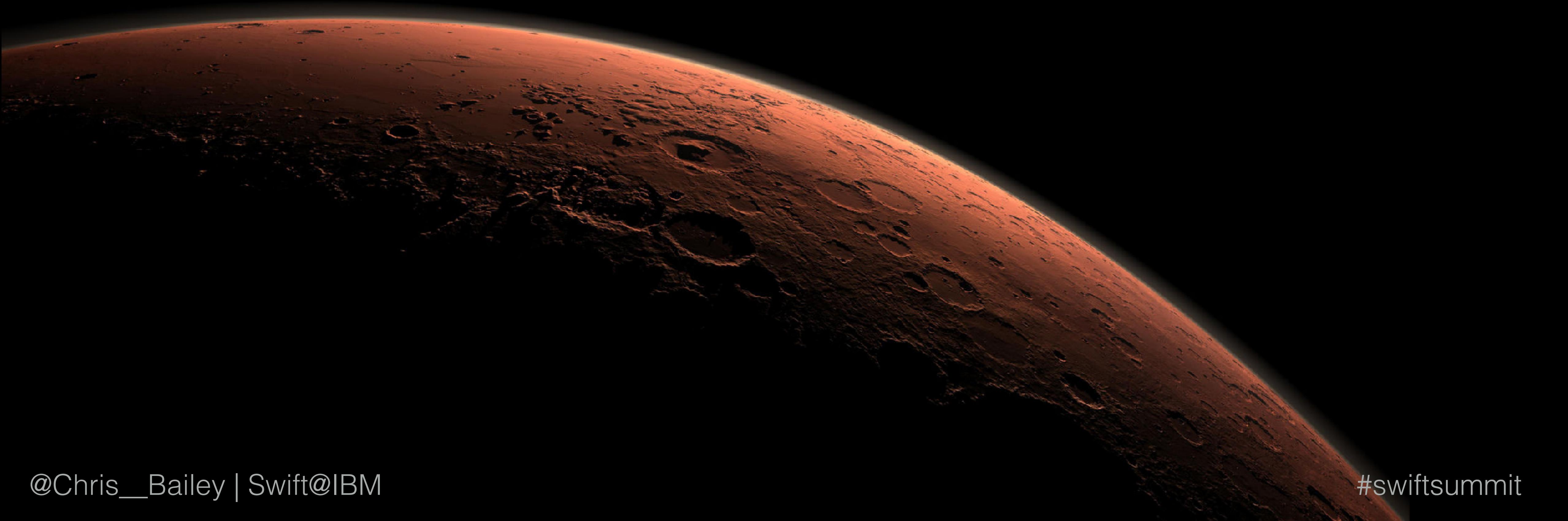
1998

MARS CLIMATE ORBITER
DEC 11: LAUNCH FROM CAPE CANAVERAL



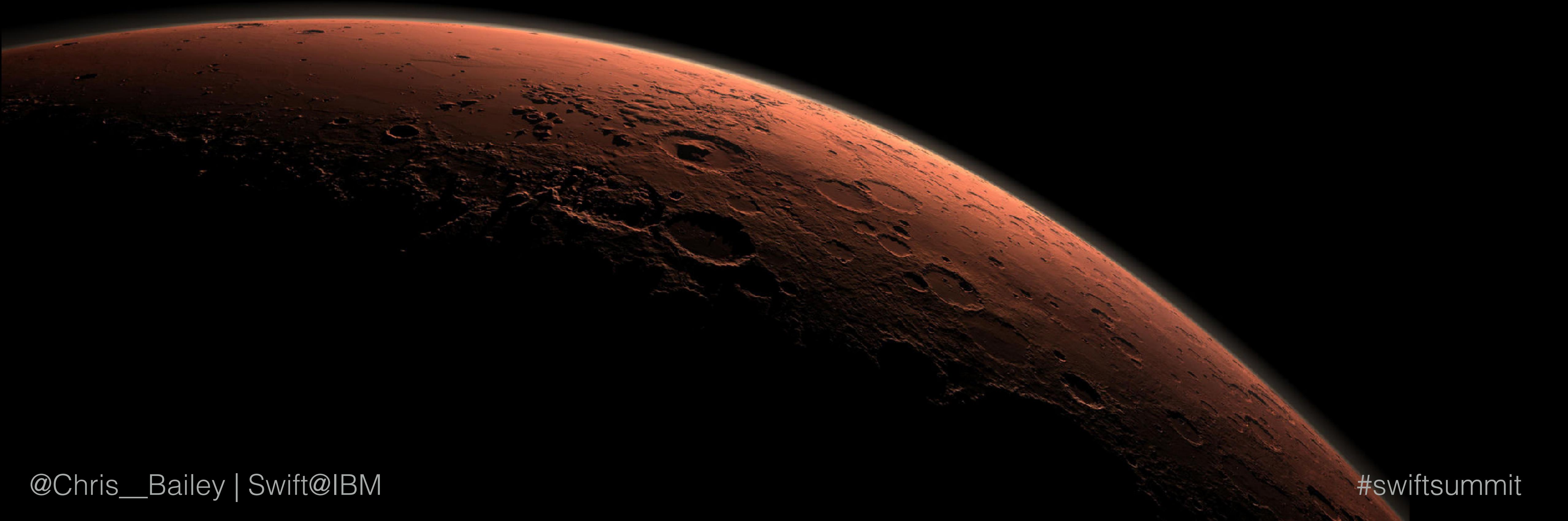
1998

MARS CLIMATE ORBITER



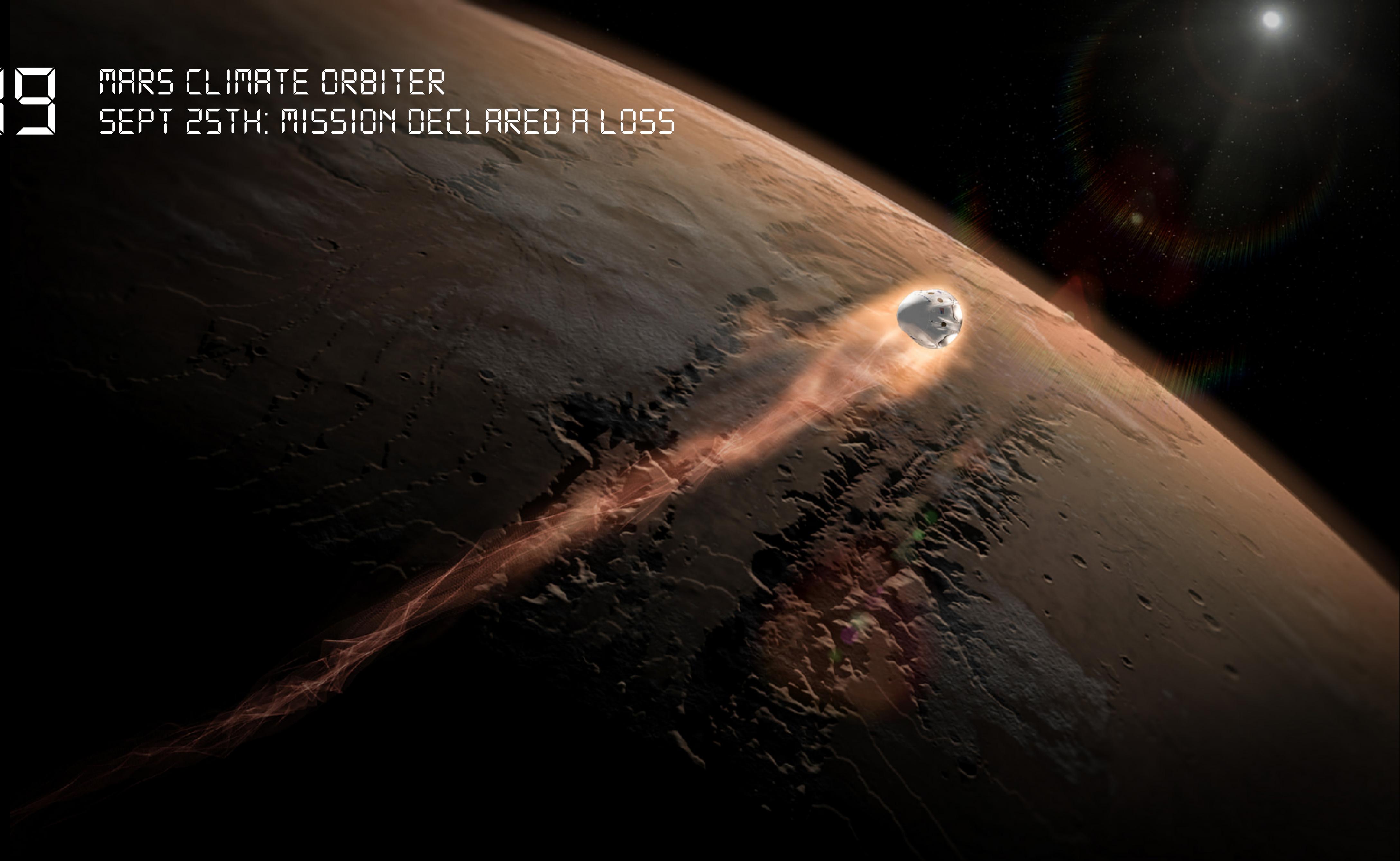
1999

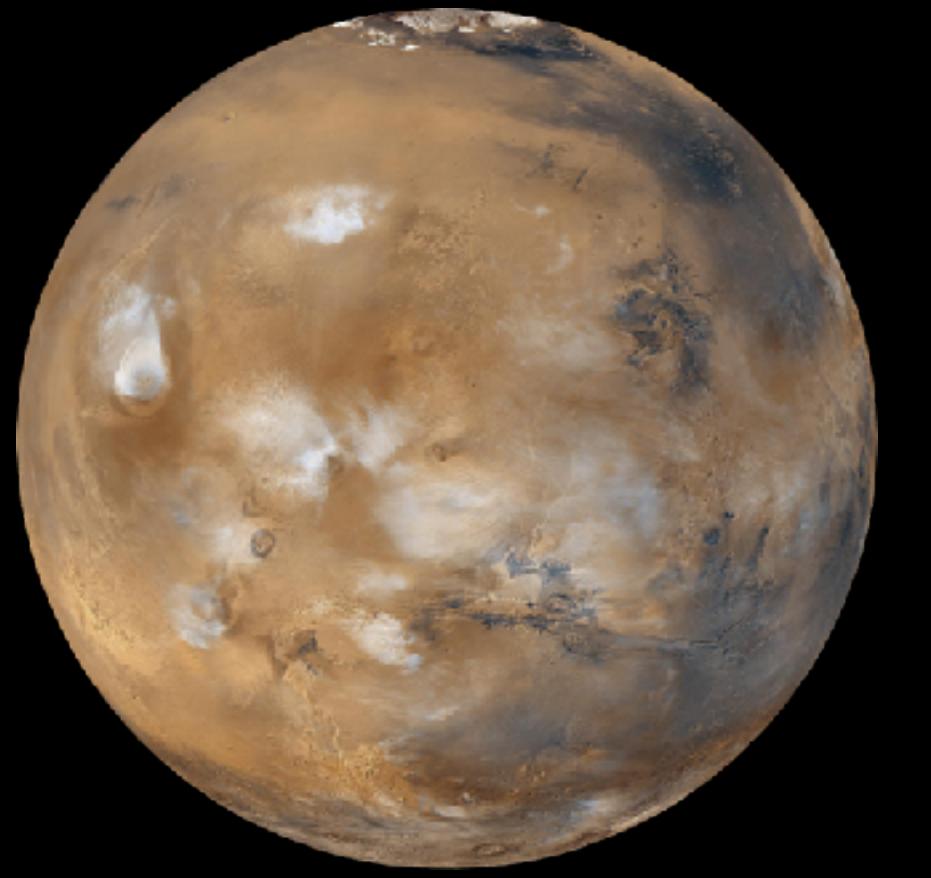
MARS CLIMATE ORBITER
SEPT 23RD: LOST RADIO CONTACT



1999

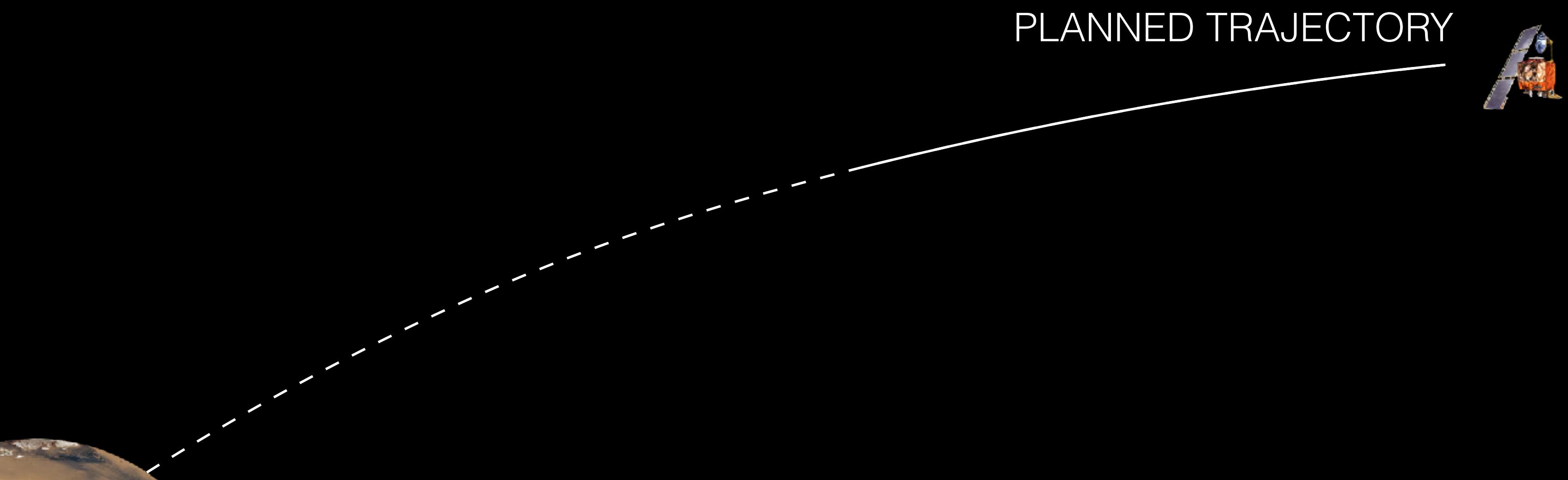
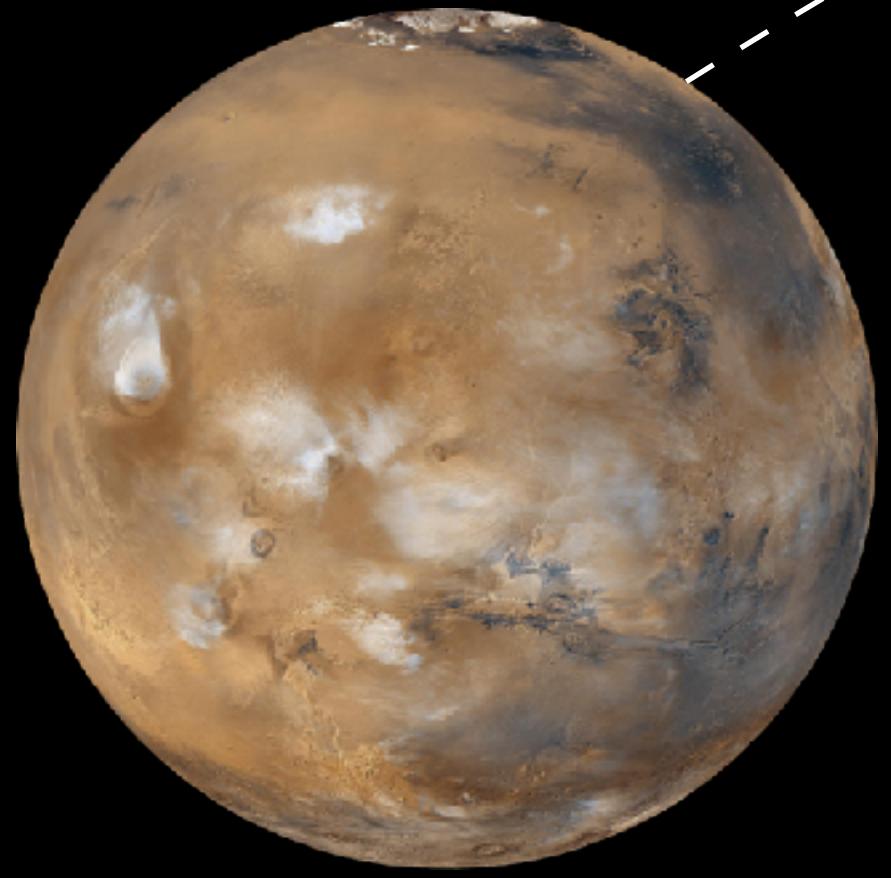
MARS CLIMATE ORBITER
SEPT 25TH: MISSION DECLARED A LOSS





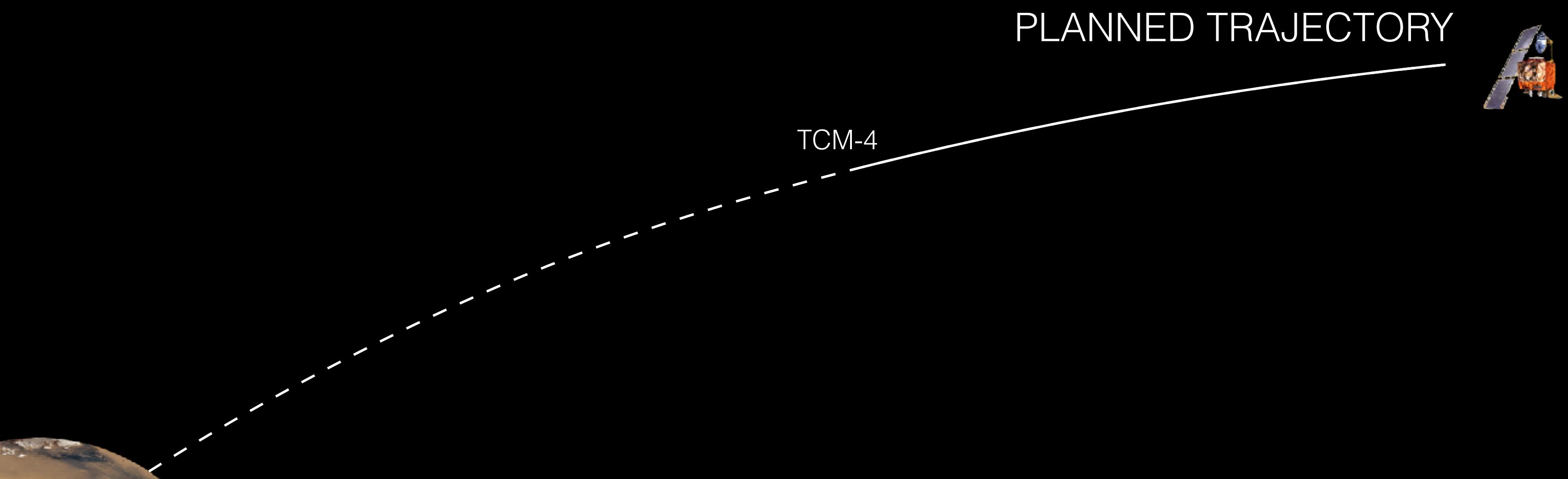
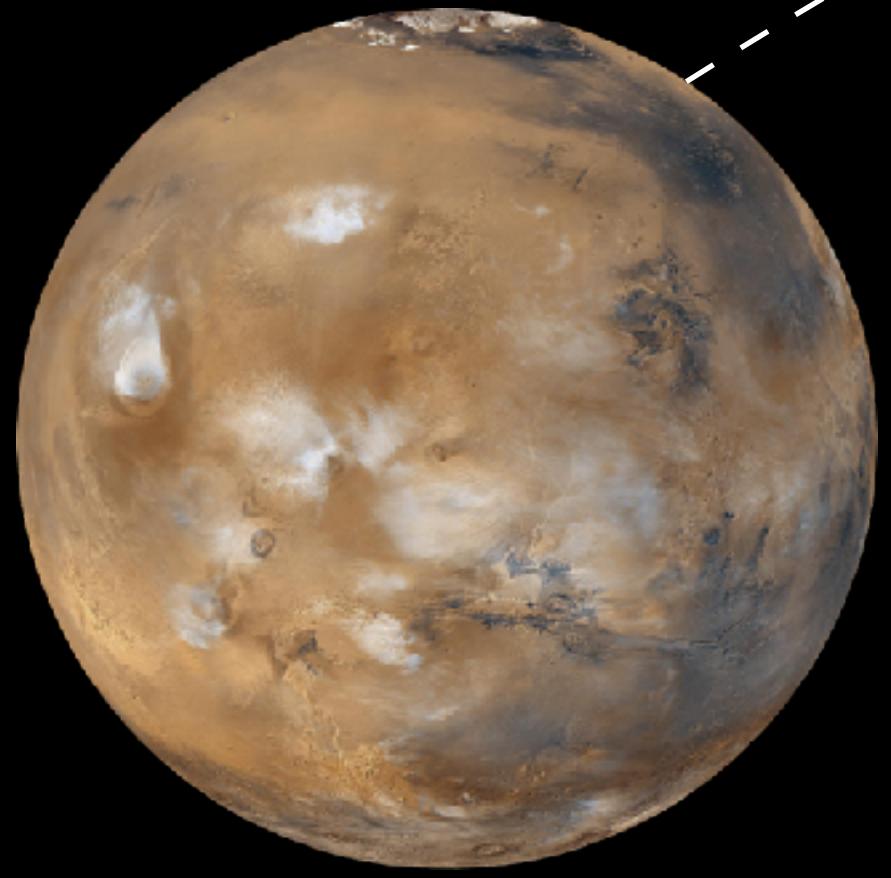
@Chris__Bailey | Swift@IBM

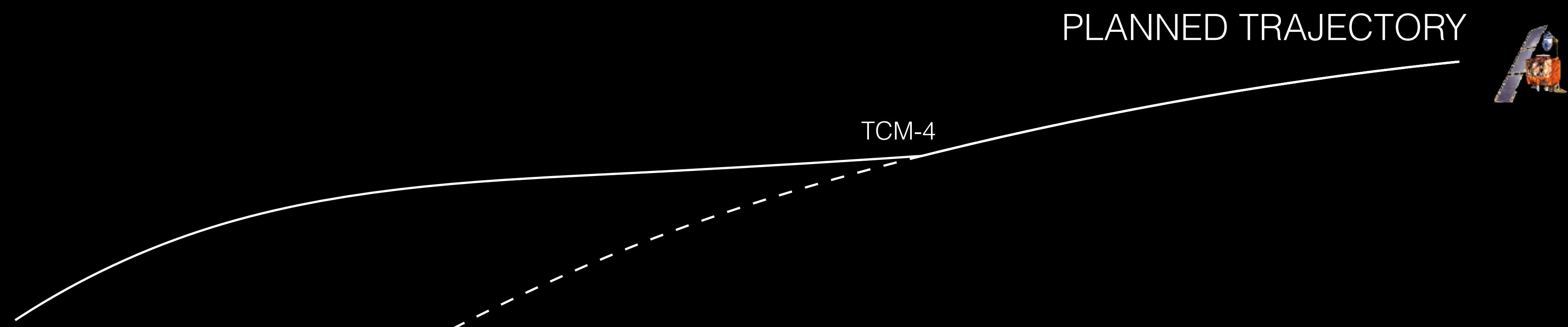
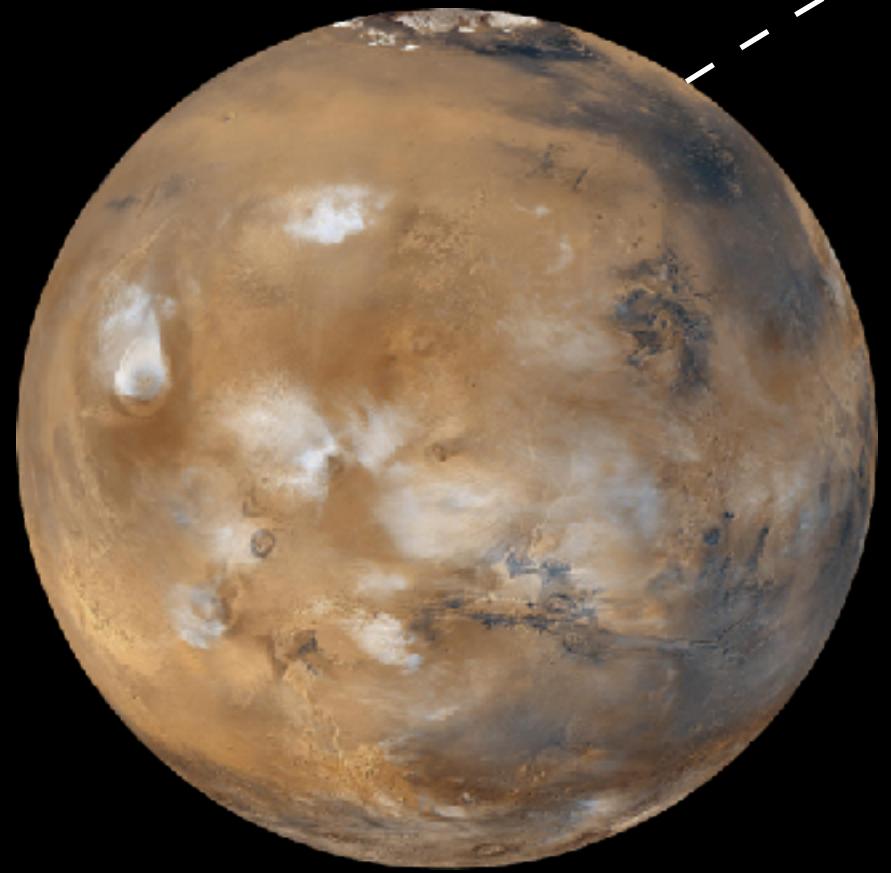
#swiftsummit

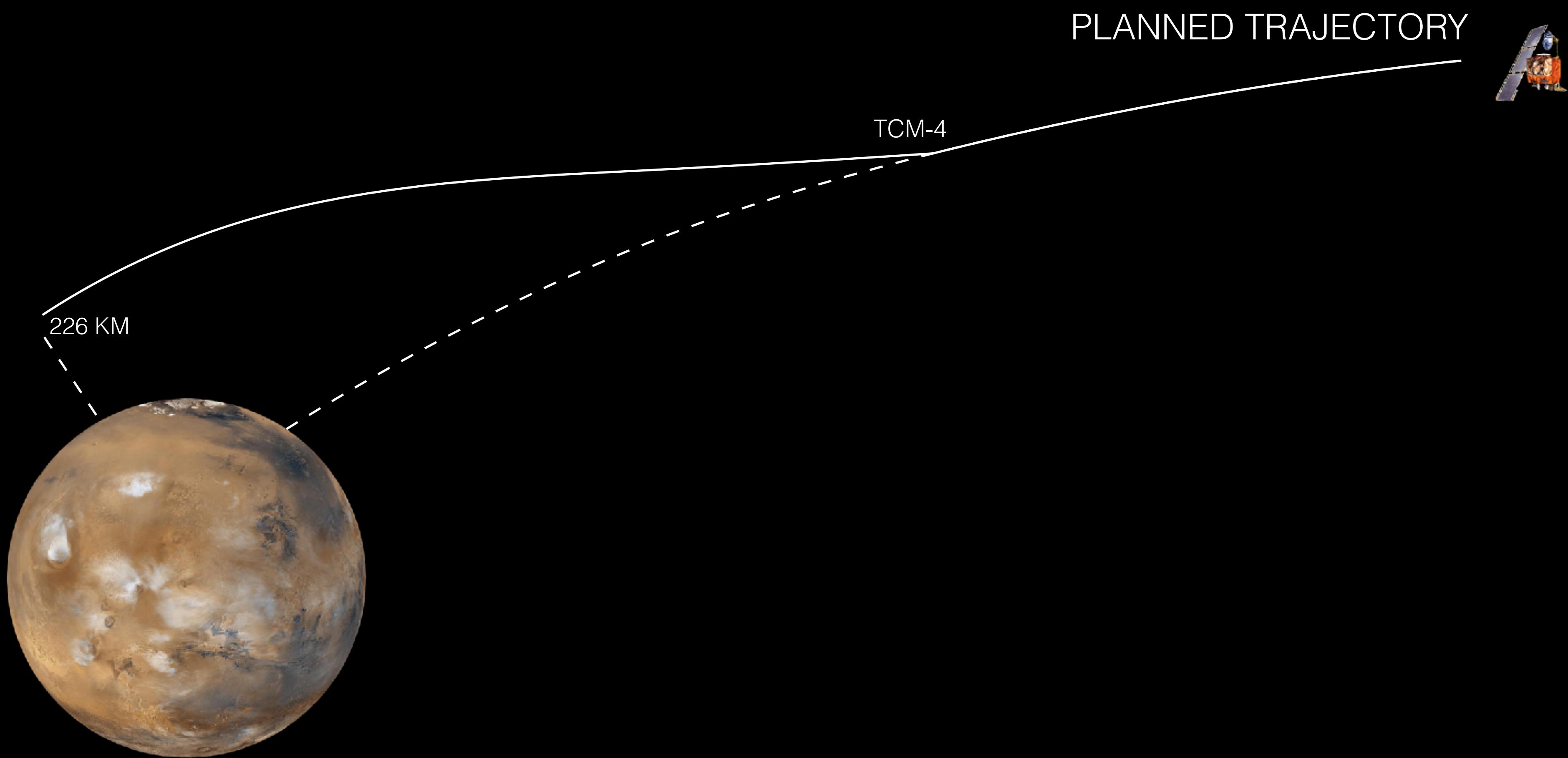


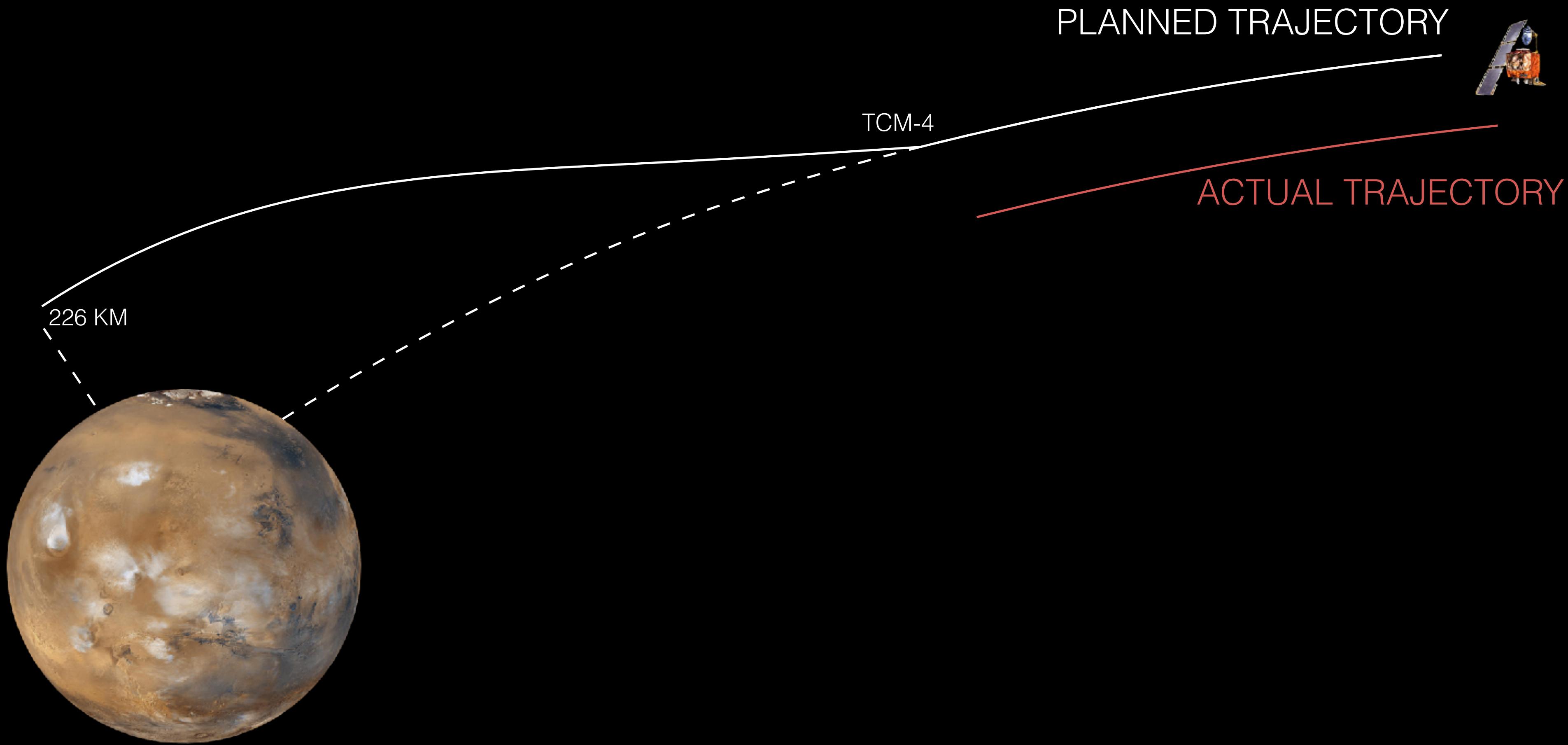
PLANNED TRAJECTORY

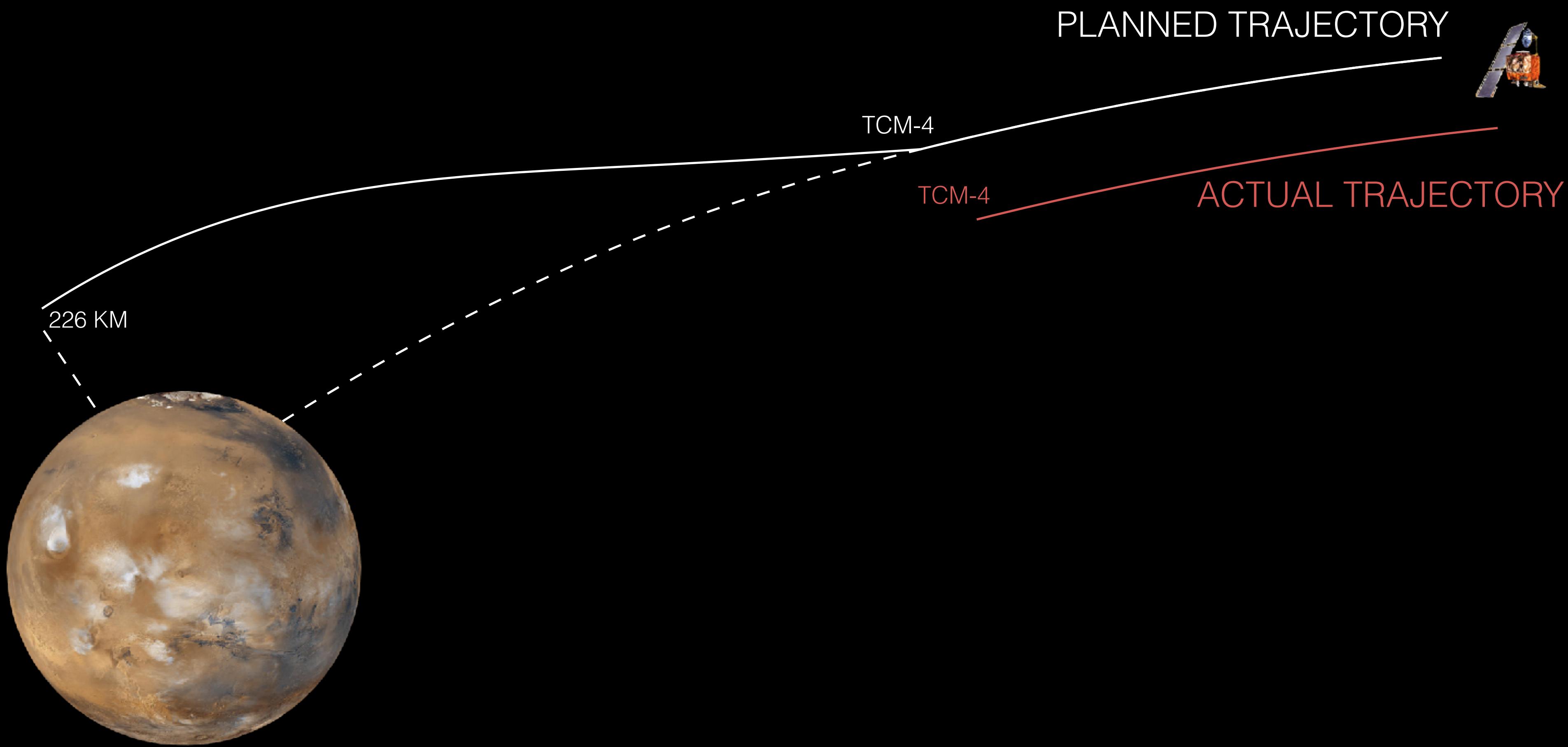


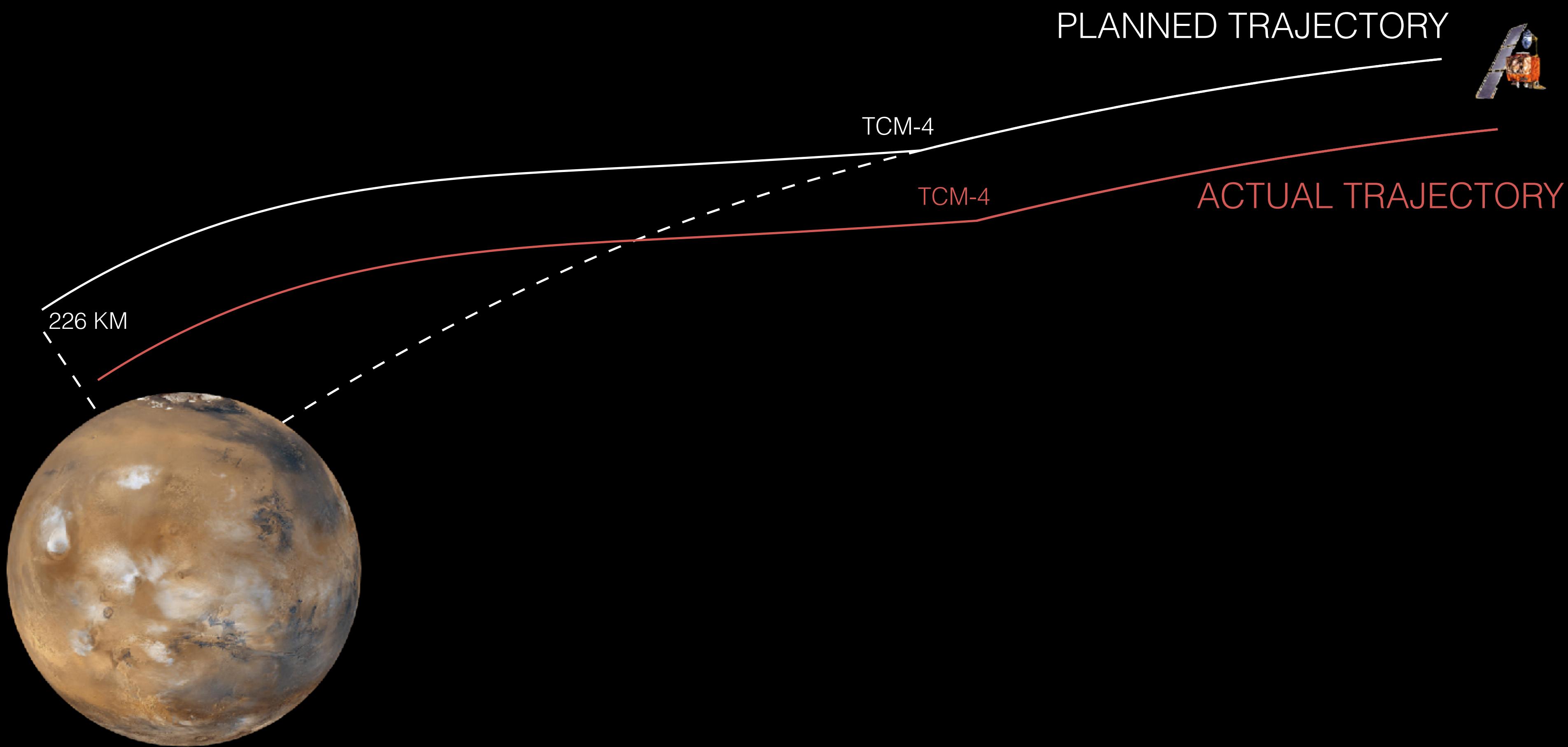


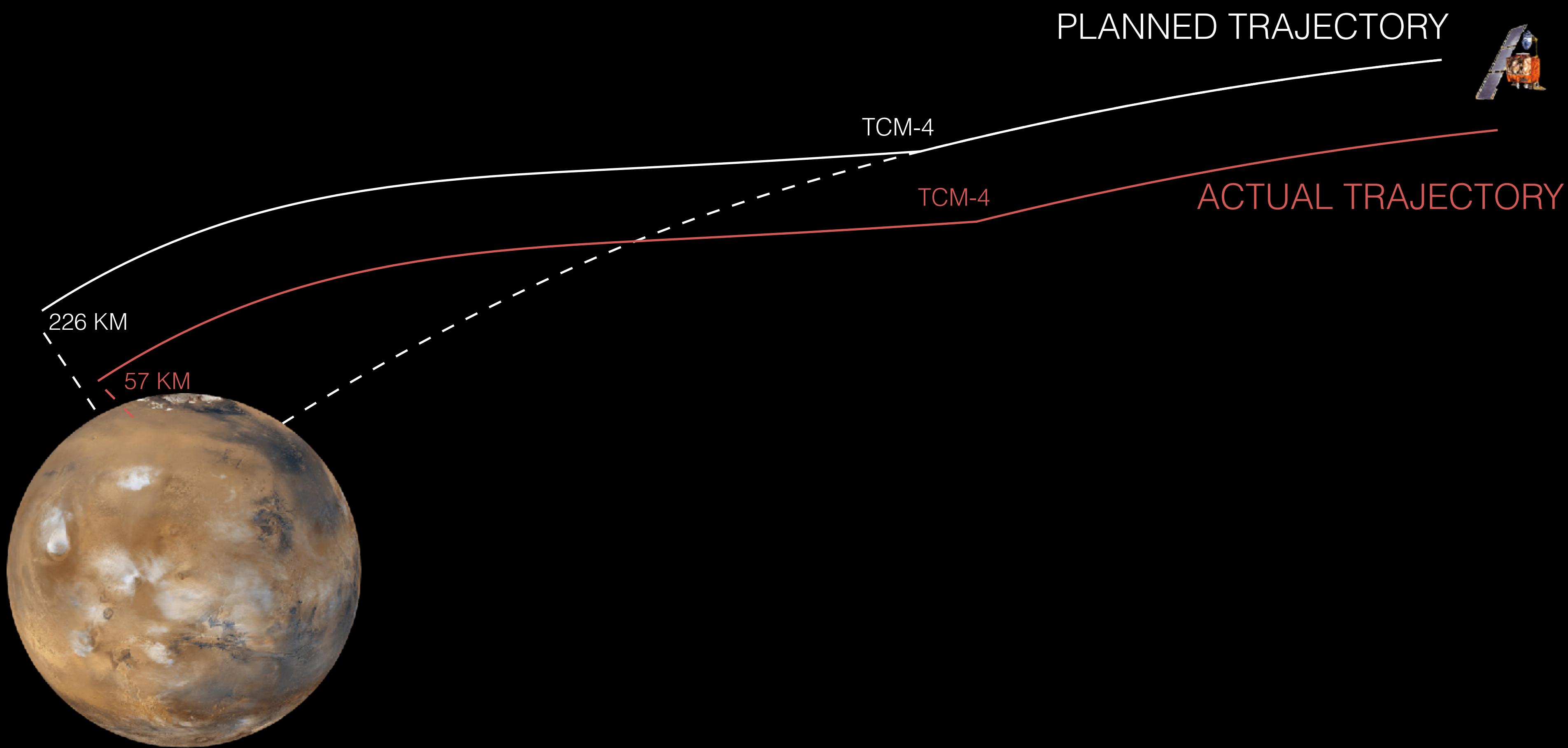


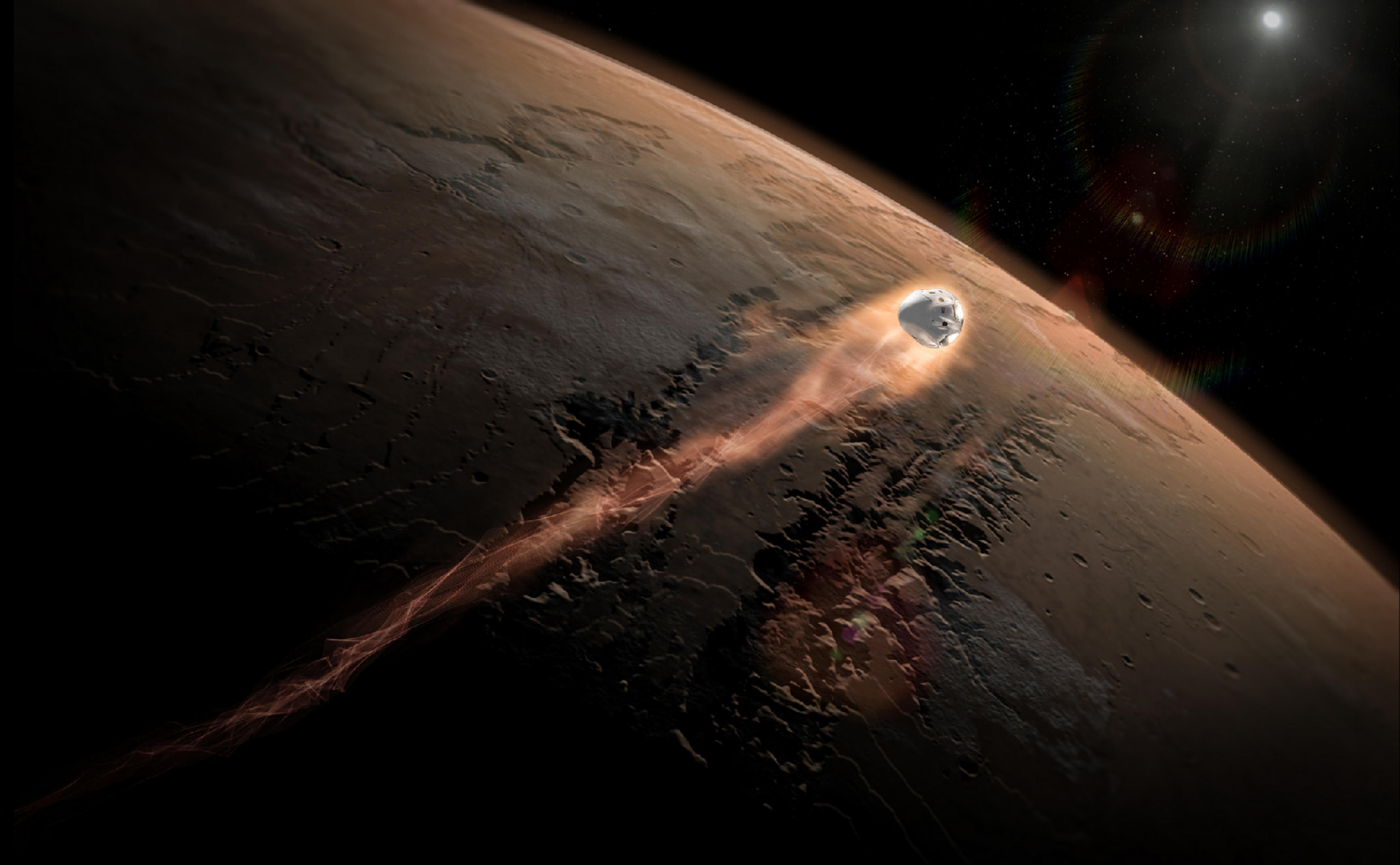












@Chris_Bailey | Swift@IBM

#swiftsummit



Nasa Jet Propulsion Laboratory

Trajectory Calculation Software

newton-seconds
(International System of Units)

Nasa Jet Propulsion Laboratory
Trajectory Calculation Software

Lockheed Martin
Ground Software

4.45

Total Impulse

1

SIS

(Software Interface Specification)

Lockheed Martin

Ground Software

pounds-seconds
(United States Customary Unit)



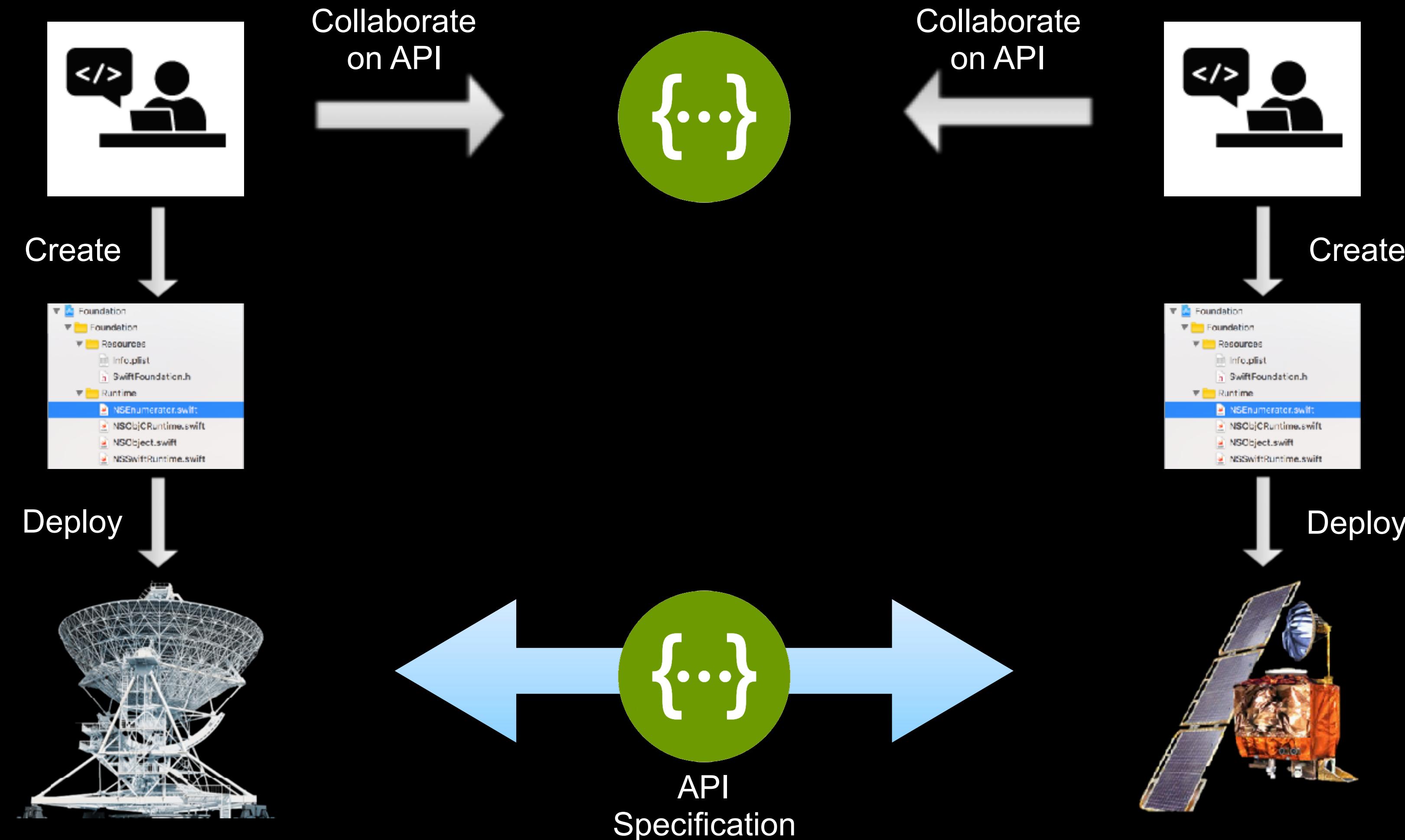
“Software interop is hard”

A dramatic photograph of a rocket launching from a dark, textured surface, likely Earth's crust. A massive, luminous orange and yellow plume erupts from the base of the rocket, casting long shadows and illuminating the surrounding area. The rocket itself is a small, white, conical shape positioned in the upper right quadrant of the frame. The background is a deep black space, with a few distant stars visible.

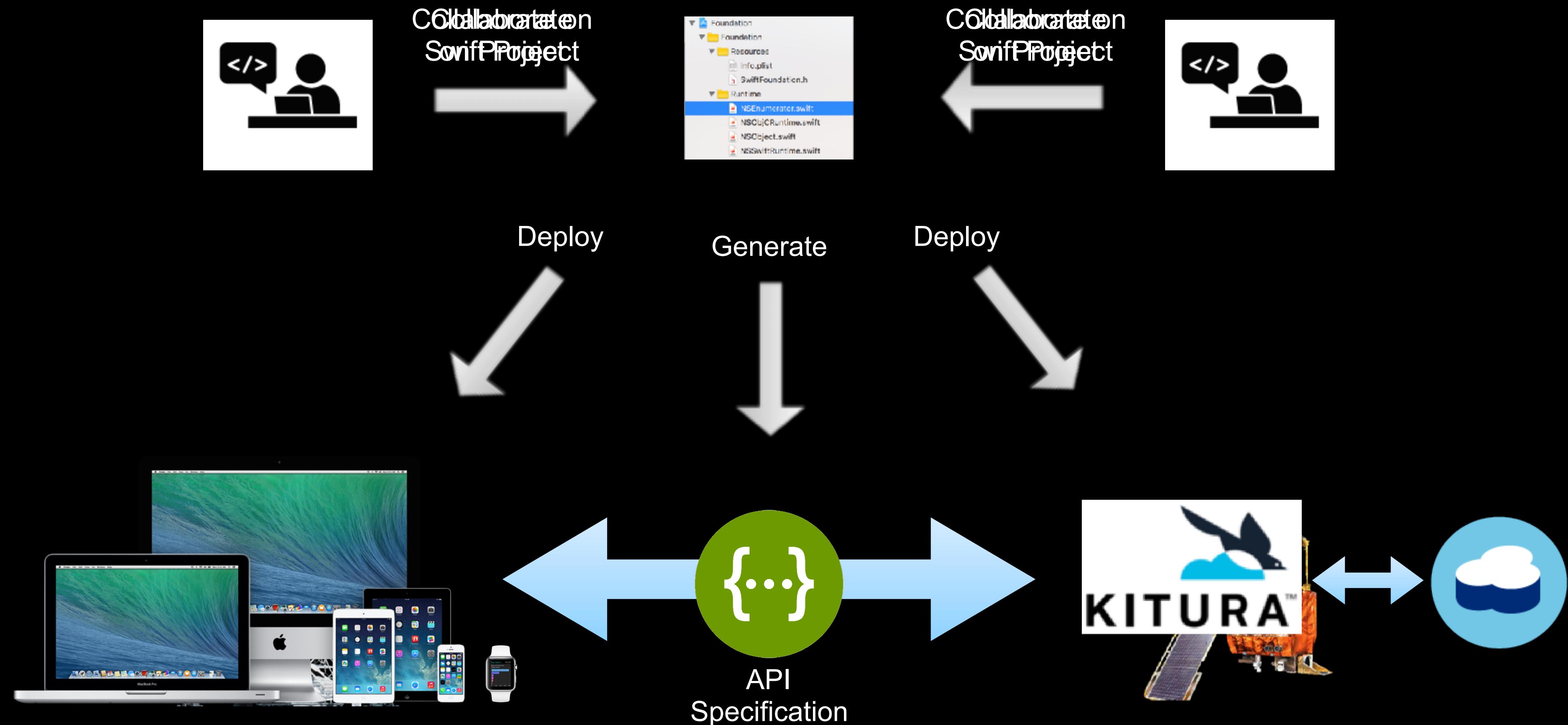
“Software interop is hard”

—Rocket Scientists

Siloed Development



Collaborative Development



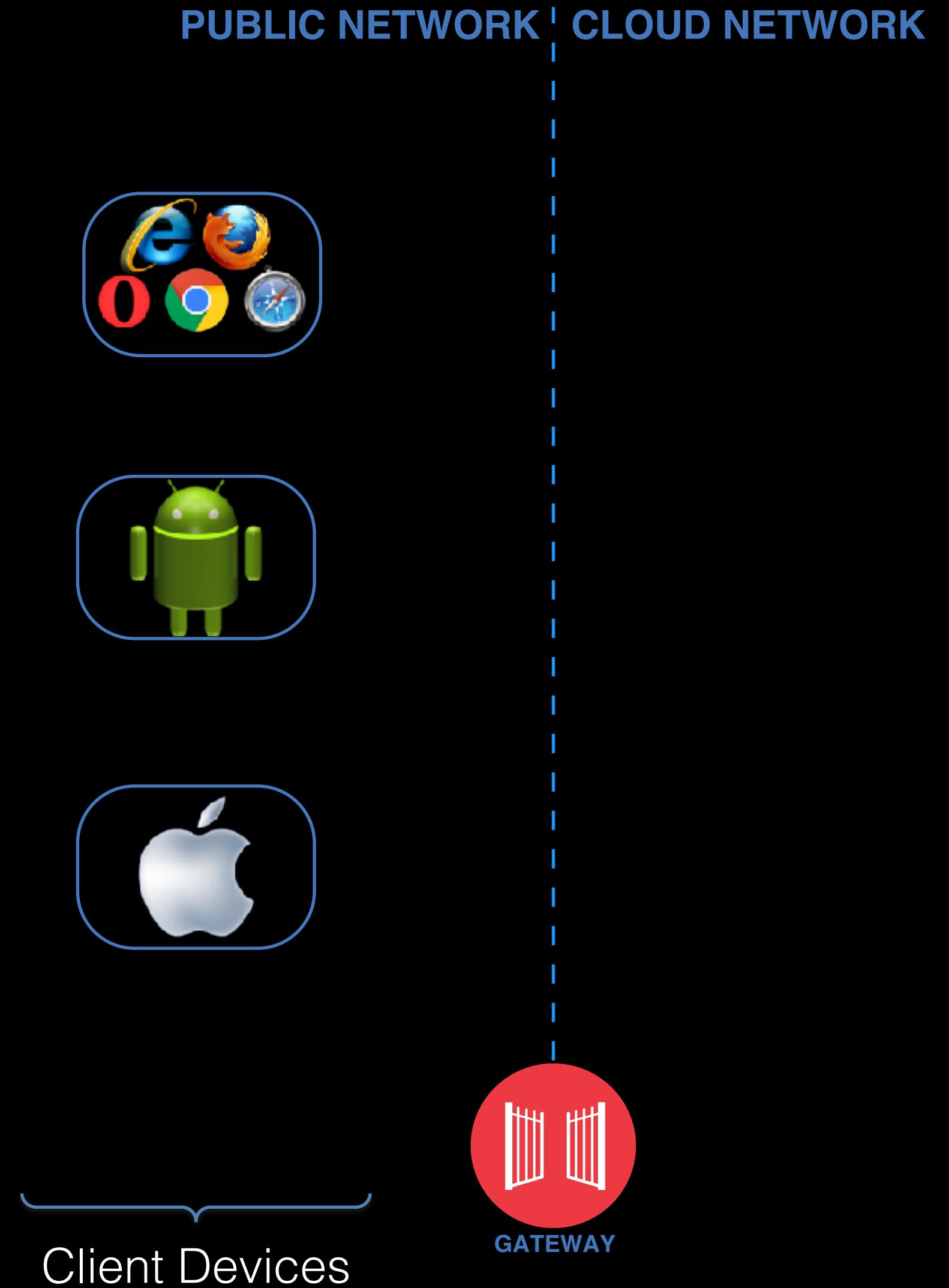
Emerging Deployment Models

Emerging Deployment Models

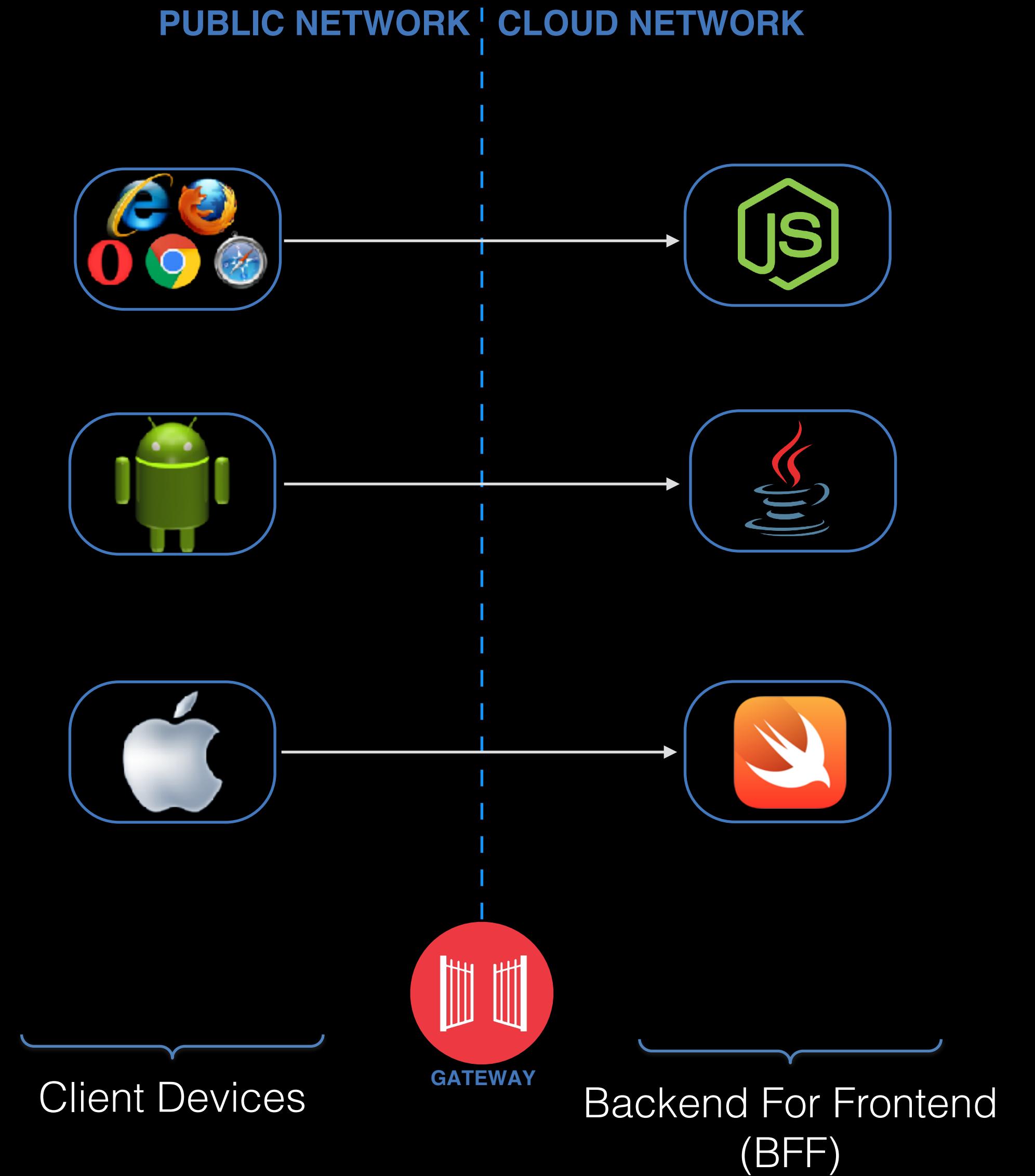


Client Devices

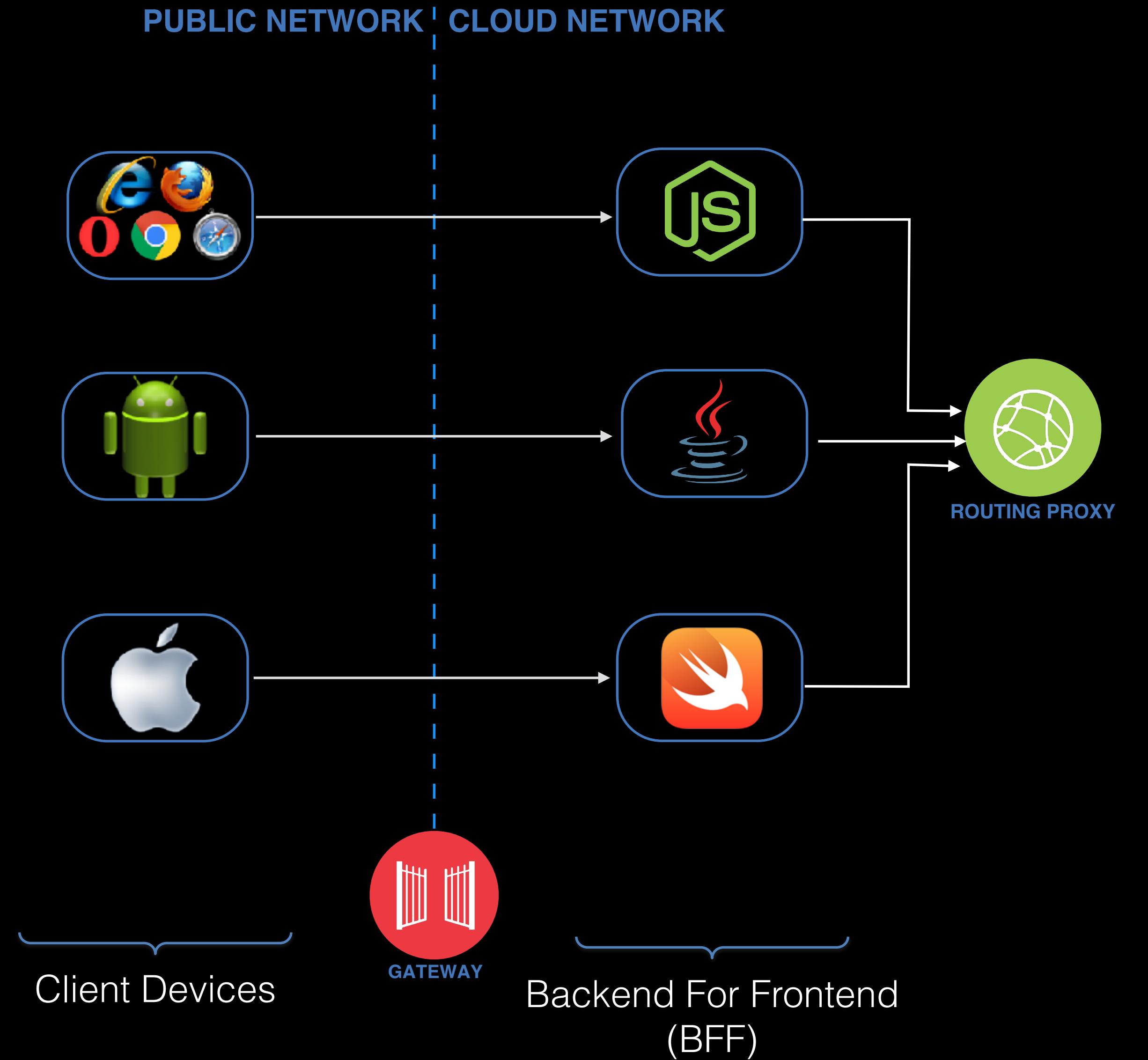
Emerging Deployment Models



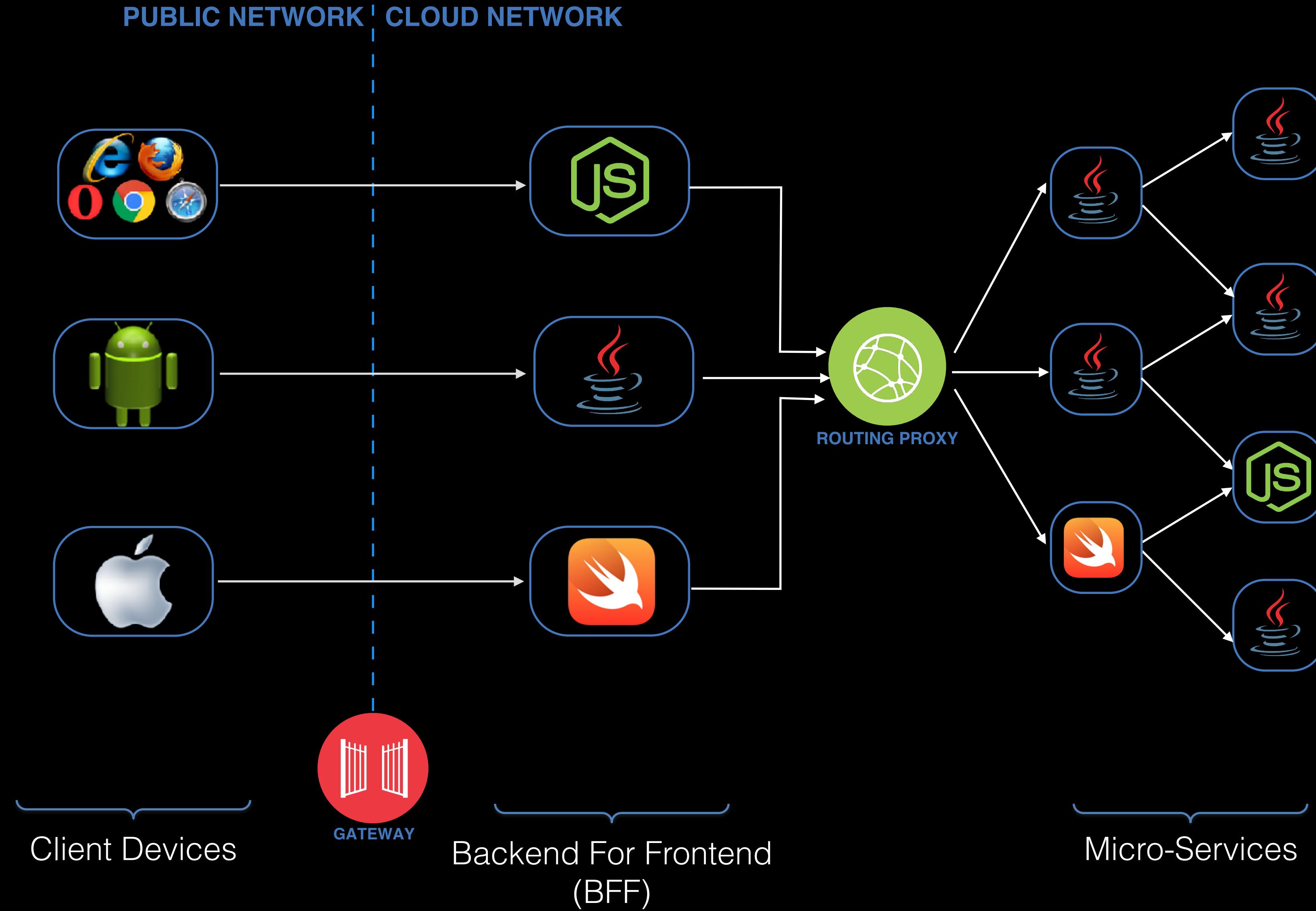
Emerging Deployment Models



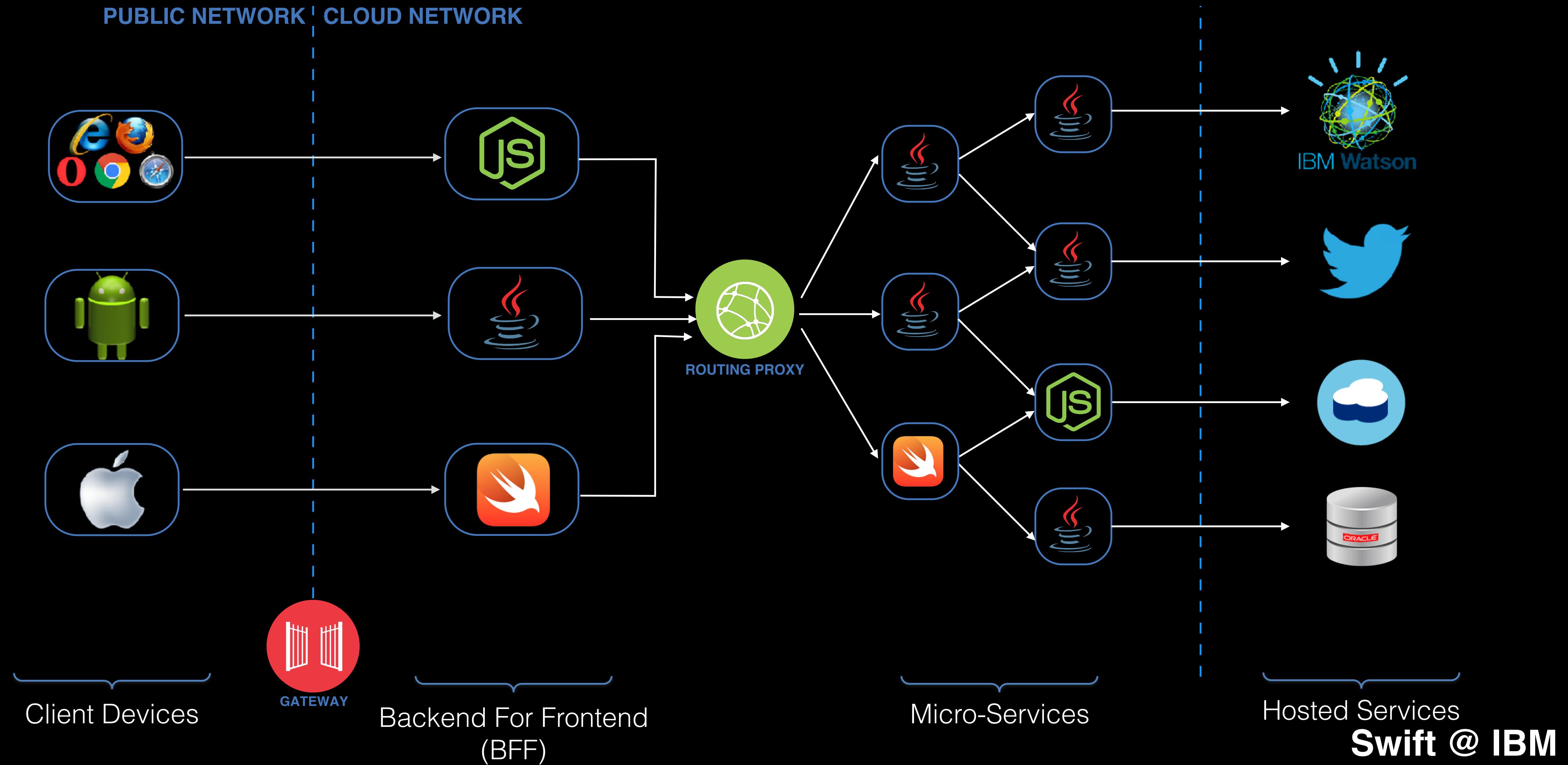
Emerging Deployment Models



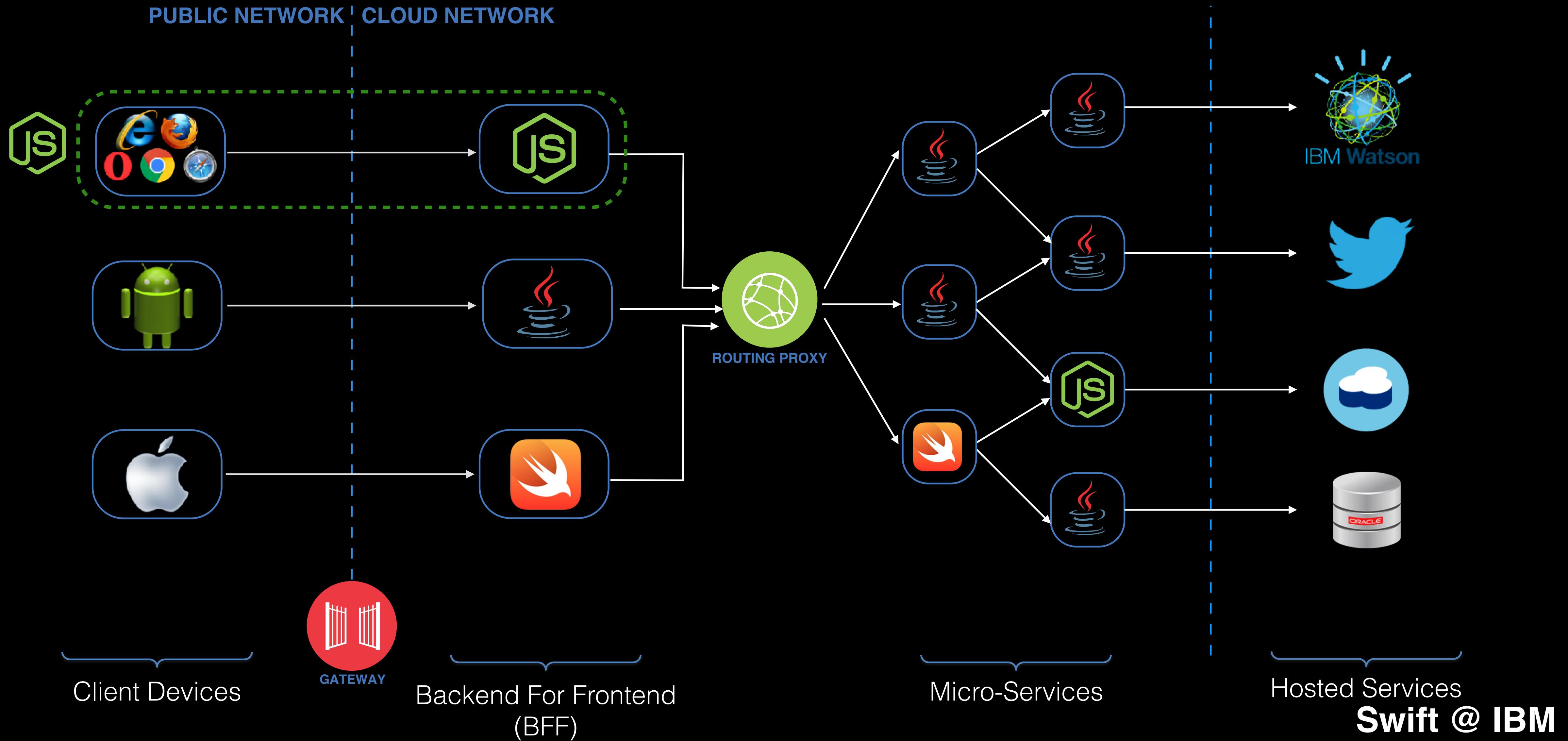
Emerging Deployment Models



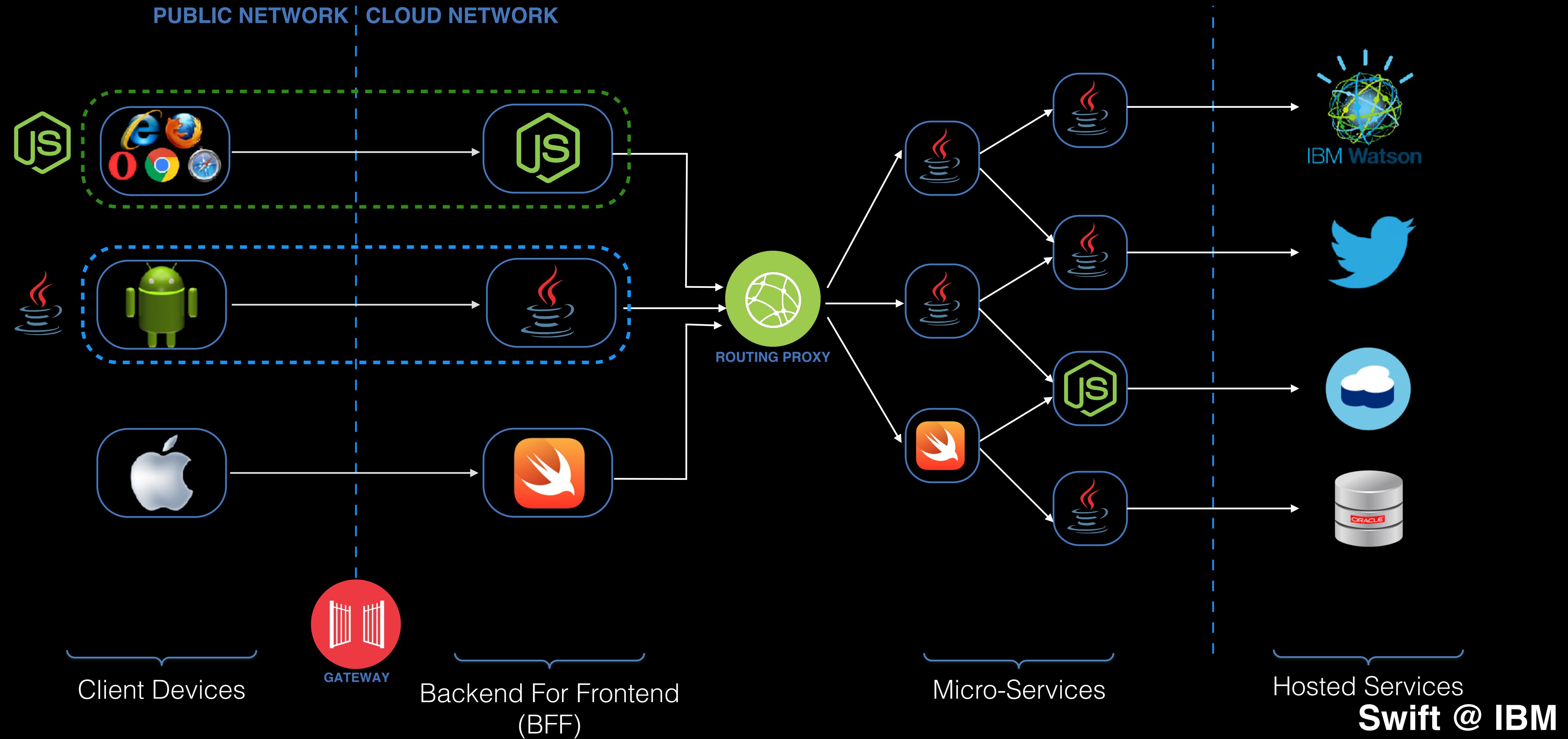
Emerging Deployment Models



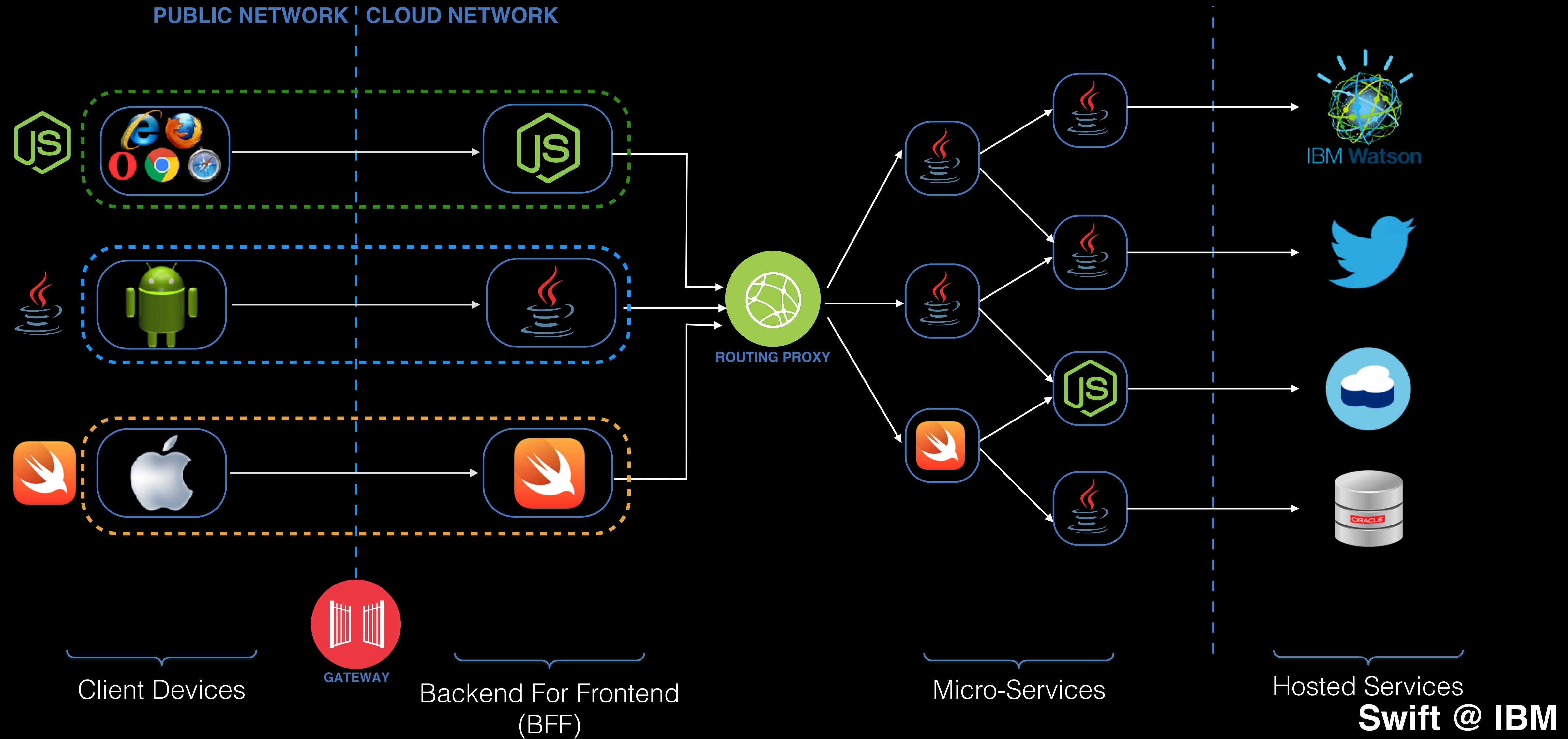
Emerging Deployment Models



Emerging Deployment Models



Emerging Deployment Models



Thank you!